Engineering Software III

Proceedings of the 3rd International Conference, Imperial College, London, England. April 1983

Editor: R.A. Adey

Volume I

Springer-Verlag Berlin Heidelberg GmbH

Engineering Software III

Proceedings of the 3rd International Conference, Imperial College, London, England. April 1983

Editor: R.A. Adey

Springer-Verlag Berlin Heidelberg GmbH 1983

R.A. Adey

Computational Mechanics Centre,
Ashurst Lodge,
Ashurst,
Southampton,
Hampshire, SO4 2AA,
UK.

ISBN 978-3-662-02337-2 ISBN 978-3-662-02335-8 (eBook)
DOI 10.1007/978-3-662-02335-8

Originally published by Computational Mechanics Centre, Southampton 1983
Softcover reprint of the hardcover 1st edition 1983

CONTENTS

C.A.E. INFORMATION, EDUCATION AND EXPERIENCE

CIVIL ENGINEERING

Preface

These proceedings contain the papers presented at the Third International Conference and Exhibition on Engineering Software held at Imperial College, London during the period April 11th - 13th, 1983. I must thank again the authors who submitted the large numbers of papers which made selection a difficult task.

The theme of the conference is the use and application of computers in engineering. Many abbreviations have been invented to describe the use of computers from CAD, CAM, CADMAT etc. but the term which best describes the scope of the conference is Computer Aided Engineering, CAE.

The papers have been split into sections covering different application areas such as Mechanical Engineering, Civil Engineering. Other sections cover techniques such as Finite Elements, Boundary Elements and General Simulation. An important session at the conference was the new field of engineering databases and as in past conferences the special sessions were devoted to microcomputers.

R.A. ADEY
(EDITOR)

ENGINEERING SOFTWARE DESIGN

MENU INPUT GENERATING SYSTEM FOR THE FORTRAN PROGRAMS

I. Kovačič

Institute of Structural and Earthquake Engineering
Department of Civil Engineering
University "Edvard Kardelj" of Ljubljana, Yugoslavia

INTRODUCTION

Although fortran is losing competition with the new languages it is still very used programming language, especially in the technical software production. Technical tasks are not to be described by a lot of data usually, as in business applications. In such a case in fortran supplemented facilities are sufficient and no troubles are to be expected. In the case of the interactive program with bigger input part more efforts must be made to produce a good user pleased input. Such an input can be based on the following principles:

- By the menu technique the user decides, which part of the input data he is going to define. The data description is then made in interactive dialogue mode. The program displays questions with descriptions of the expected data. In the case of errors the message must be displayed and correction must be enabled.
- It must be possible in every moment to see the state of the data. If the changes are needed, this must be possible with the minimal effort.
- The format free input in the dialogue mode is a rule, but it must not be limited to numerical data. The writting of the word in quotes only is not very acceptable solution. In fortran implemented format free input-output statements do not satisfy these demands.
- The selective help information must be available providing the user to act quickly and properly when in troubles without searching the manual.
- The same program must work in the batch mode too. The dialogue must be changed by talking and suitable action must be provided in the case of errors.

All these principles are independent of the problem of the application program. The experienced programmer can only provide this sort of input. One way to get cheaper solution is to make

suitable software tool once and use it many times. This could be a program, which accepts the description of the input as data, and generates automatically the complete input part of the application program. Many such tools are made, but except program "CADINT", which is designed for batch environment primarily, they are not available here. As a part of two dimensional general finite element project MIGS - Menu Input Generating System - was designed and realized.

General characteristics of the system are described in this paper.

MIGS - MENU INPUT GENERATING SYSTEM - GENERAL DESCRIPTION

Basic principles relating to the design

Besides the principles presented in the introduction some additional requests were set up in the design of MIGS.

- The most important factor of the program is the final user. The pleasant way of presenting data to computer is one of the guarantees that the author of the program is not the only user. The displaying redundant information can slow up the response time, therefore the dialogue must be simple and concise. It is easy enough, for example, that the key "?" means a help request in the current situation. There is no need to display menu over and over, because the user will keep it in mind soon, but he must have the possibility to request it again with a simple command any time. The experienced user, who knows the data structure in detail, will not allow the program to put him questions about data, he will simply tell the data to the program and that can be much faster. The structure of the data must not be limited to simple constructions. Sometimes the alternatives in presenting data can enable more natural way of communication between a man and a computer.
- The input generating system must be useful for small programs too. In that case everything must be simple and economical: input generation, programming of the application program in which the generated input is included and the final usage of the program. The special solutions of the system must be separated from the standard ones. There must exist the possibility to use only the menu generating part of the system or the data language definition part or both. The design of the system must enable simplifications and extensions.
- Except some good documented nonstandard elements the whole system must be written in the standard fortran IV. The encode/decode and the file manipulation facility are such exceptions. All storage consuming definitions must be parametrized to enable quick and correct implementation to the chosen computer.

General external characteristics of the system

MIGS was designed in the harmony with all requested principles. It was coded in the fortran preprocessor named Structran, good enough to produce well structured and parametrized programs. MIGS consists of the subroutine library "MIGLB" and the main programs: MIGDF, MIGGE and MIGRE.

MIGDF This program is intended for the menu definition and the data language definition. It can be used in the dialogue or in the batch mode. The resulted information is written to the sequential file in the compact binary (compiled) form. MIGDF is actually a sort of an editor, because it enables changes and additions to the existing information. When executing in the dialogue mode, the complete user part of the dialogue is written to a log file, which can be used later as the input for the same program in the batch mode. It is clear that we can edit this source file with the standard text editor. There is no doubt that MIGDF uses the MIGS system itself for the input. The special good facility of the MIGDF is the possibility to check the defined part of the input at once. It is enough to command "TEST" and the program is ready to accept the just now defined application program input performing the complete recognition analysis.

MIGGE This main program generates dummy processor modules of the application program, based on MIGDF output.

The resulted fortran modules must be compiled and together with the output of MIGDF and several modules from the library MIGLB represents the application program, which is able to run. The final product is obtained by changing dummy modules by the actual ones.

MIGRE It is not very comfortable to take care of different forms of the same thing. The compiled form of the input information is the one used in the application program. The main program MIGRE is able to transform this compiled form into the source one. This is also useful for the documentation purposes.

Input processing in the application program The input process is started simply by calling the module MIGRD (unit), which reads the compiled menu and the data language information from "unit" to the memory (labelled common). Then the input processor module MIGPR (menu) is called, where menu is the number of the first menu, which will be displayed immediatelly. Menu = 0 means processing without the starting menu. MIGPR reads statements and every time when a new statement is read without errors, the executing module MIGEX (stat) is called. Stat is the sequence number of the statement. MIGEX can be generated by the program MIGGE or must be supplemented as a part of the application program. The data of the read statement are stored in the labelled common /MIGVAL/, where MIGEX can find and use them.

Alternative usage of the MIGS The menu and the data language definition can be integrated in the application program, calling the proper modules from the library MIGLB, so the sequential file with the compiled input information can be avoided. This is acceptable solution for small inputs, as also for big programs with the overlay structure, where one overlay is intended for the data and the language definition.

Usage of the MIGS in small memory invirоment There is no need to store complete data and language information in the memory. The current menu's statements are the only needed in the memory. When moving to the another menu, new information must be read into the memory from the specified sequential file.

SYSTEM DESIGN OF MIGS

Data input language description
The problem oriented language must be defined considering two goals:
- It must provide a good communication tool between the user and the program.
- The recognition and interpreting data statements written in this language must be easy to implement for the programmer and efficient for the computer processor.

The syntax of the language must be in accordance with these requests. The syntactical rules of the language used in MIGS can be defined partially in the BNF notation (Katzan 1970):

```
<statement>::=<statement name><end of statement>/
              <stat. name><blank delimiter><data><end of stat.>
<data>     ::=<datum>/<datum><datum delimiter><datum>
<datum>    ::=<value>/<datum name><name delimiter><value>
<value>    ::=<nul>/<integer>/<real>/<double>/<word>/
              <literal>/<scalar>/<vector>/<record>
```

Names "Statement name" or "datum name" is any string starting by the alphanumeric part with first letter character. Any special character (blank also) stops this part of the name. At the definition stage the minimal recognition length of the alphanumeric part must be provided. Only the alphanumeric part is used at the reference level. For example the name "Start point (x,y)" with the minimal recognition length of three characters can be referenced only by the following strings: "Sta", "Star" and "Start".

Delimiters All delimiters are defined special characters, which can be preceded or succeded by zero to any number of blanks. The default characters of delimiters are: blank for the "blank delimiter", "," or blank for the "datum delimiter", "=" or blank for the "name delimiter", ";" or "!" or return key for "end of statement". All delimiters can be redefined by other characters. The maximum number of the different delimiters of the same type is the system parameter; its value is usually 3.

Example of two valid statements:

```
LINE FROM 2,3 TO 0,0 TYPE=solid, COLOR=green; PLOT! Comment
```

The name of the first statement is "LINE", it has four datums: "FROM", "TO", "TYPE" and "COLOR". The values of the first two are vectors, the last two are scalars. The second statement is without data. All characters to the right of ! are not the part

of the statement, their function is to comment the statement on the left.

Order of datums It is possible to write a statement without the datum names. The order of the datum values must agree in that case with the defined order of datums. If the datum names are used, any order of datums is possible. If names are omitted in the middle of the statement the sequential datum order is assumed:

```
LINE COLOR=green FROM 2,3 0,0 solid
```

The order of the datums can be defined very generally: for every datum its successor can be defined. This option is useful in connection with the another option which enables to define alternate datums. For every datum an alternate datum can be specified. Both facilities enable to define nonlinear data structure. It is altogether intended just for special purposes.

Long statements The end of statement characters enables writing more statements in one row. A special character _ is reserved to continue one statement over more rows:

```
LINE 2,3 0,0 solid green:, TEXT START=5,5 FI=90 STRING="press_
ure", COLOR=yellow; PLOT
```

It is possible to define the statement as "long". Such a statement can be finished by ";" or "!" only. It can be spread over several rows without using the special continuation character "_".

Repeating the same statement When repeating the statement without writing the name again, old values of undefined datums can be used. It depends of the application program what to do in this case. It is easy to inquire the state of the value of any datum of the current statement: defined, undefined, or old statement value. If we repeat the statement name too, all old values are cleared before reading new data.

Dialogue When typing the name of the statement alone following by the return key, the program responds displaying the list of datum names (in full length) expecting to read their values:

```
LINE <return>
FROM TO TYPE COLOR
```

The same rules as in the non-dialogue mode are valid in this situation. The answer followed by new statements can be written in the same row. The dialogue mode is switched off in the batch enviroment. The log file where the user input is copied can be defined in the dialogue and batch mode.

Error processing In the dialogue mode the row or rows where the statement in error takes place is displayed again together with an arrow in the next row showing the error location followed by the short written message. In the batch mode the error message is written to the log file. The statements to the right of the statement in error are discarded.

Statement definition The statement sequence number is the only unique identifier of the statement. It is possible to have statements with the same names but not the same sequence numbers. The same is true for datums where the following information is to be defined: datum sequence number, datum name and datum value type number. The separate modules are provided for the statement and datum definition, enabling changes of a single datum.

Data types
The set of possible values of the datum is determined by the type of the datum. The only types allowed in the system are those chosen from the prepared classes of types: integer, real, double, word, literal, scalar, special character, vector and record. For the new type the class and the parameters which determine the unique choice, are to be specified. For example, the literal is defined by: limit number of allowed characters, maximum number of requested characters, start and end special characters (can be different) and information about changing small case letters to the upper automatically.

Integer, real, double They are types known in fortran. The parameters define the picture of the values. For example, the field length and the number of decimal places must be specified in the case of real.

Word The values of this type are the strings of characters without blanks. The limit and maximum requested number of characters, information about upper case letters are parameters of this class. With special request it is possible to forbid special characters in the word (alphanumeric word).

Scalar The set of the allowed scalar values of the type "word" must be defined. We have to define "word" type and number of scalar values first, then the scalar values. Every scalar has its sequence number. The minimal recognition number of characters must be given too.

Special character One special character is the value of this type. It is possible to allow any special character or determine just one.

Vector and record The set of elements of specified type defined former is accepted vector or record type. The vector elements are all of the same type, the record elements can be of different types. The vector elements can be of the record type too and vice versa, but the recursive definition is not allowed

directly or indirectly. The maximum number of elements and delimiters between the elements must be given. It is possible to request the fixed (maximum) number of elements or variable (less or equal maximum) number of elements. The start and the end special character can be specified. For example:

```
[1,2,red 2,3,blue 2,5,green, ... 5,6,red]
```

is the vector of variable length, whose elements are the records of two integers and one scalar separated by commas. The delimiter between vector elements is one or more blank characters. The vector elements are enclosed in brackets.

Literal If the start and the end special characters are blanks we obtain literal without "quotes". In this case "datum delimiter" or "end of statement" are not allowed characters in the literal, because they stop the literal in such a case.

Type definition For every class the special type definition module is included in the library MIGLB . The type definition of certain class consists of the type sequence number and of the parameters, which determine the chosen type from the class. The type sequence number is the only type data used at the statement datum definition. It is possible to change the type of the datum during the execution of the application program.

Alternative types There are situations where the choice of the type of the datum is permitted to the user. For example, if working at the graphical terminal the user can type x and y coordinates of the point or he can show it on the screen moving the crosshair. The crosshair can be started by typing the determined special character, let say "#". To define the type with alternatives, it is enough to add a list of the alternative types to it. The alternatives are tried in the same order as defined. Only the alternative types with different structure are meaningful.

Communication between the input and the application program The labelled common /MIGVAL/ represents the communication area, where all read information of any single statement is stored. The order in which values are stored is defined by the sequence numbers of the datums. The maximum length of every datum type is known in advance. Adding these length together the maximum statement storage requirements can be calculated. The first integer element in /MIGVAL/ determines actual maximum length of /MIGVAL/. The size must be big enough to accept the longest statement values. If every statement has its own execution module, it is possible to define /MIGVAL/ specially. For example:

```
INTEGER  type, color
COMMON  /MIGVAL/  length, xa,ya, xb,yb, type, color
```

in the case of the LINE statement. For the scalar we obtain the resulted scalar sequence number in the integers "type" and

"color". The special modules are available to ask for the scalar names.

Menu definition and processing

General The purpose of the menu is to limit the choice of the action in such a way that the list of possible actions can be presented to the user, who decides for one of them. In MIGS the word "action" must be replaced by the word statement. The list of full names of statements is seen on the screen. The user types the name, usually shortened, together with the data, or just the name if prefering the dialogue mode. Instead of the repeating display of the menu afterwards, only the prompt character ">" appears. If it is desirable to see the menu again, the special character "*" has a task to recall it on the screen.

Menu definition When defining menu, the menu sequence number and the existing statement number assigned to the menu must be given. The statement name of the assigned statement becomes the menu name too. Usually this is a statement without data, but it is not the rule. Every menu list element consists of the sequence number of the element, and the statement number.

Menu processing The menu is started simply by typing the statement name assigned to the menu together with the data if there are any. The assigned statement is processed first, the menu is displayed afterwards.

Menu structure There are two different solutions how to work with several menus. If the statement from the list of a menu is the one assigned to the another menu, the vertical menu structure is achieved. This device enables to define not only the tree of menus, but any connected graf of menus is possible. Another type of moving between menus is provided. To every menu its predecessor and successor can be defined. Special character ">" ("<") takes care of going from current to the next (former) menu. This enables linear horizontal structure of menus. The combination of both options is also possible. In the case that the predecessor and the successor are not defined, the automatic dynamic definition of predecessors and successors during menu processing is present. In that case "<" means to go back to the menu where we come from.

Global access The direct global access of any statement can be defined. It means, in other words that such a statement is executable any time although not present in the menu list. If this is the menu statement, the transition from current to that menu is achieved.

Help facility A short help information of the every statement can be incorporated into the input at the definition stage. The special characters "?" is used for recalling this information to the screen. Typing just "?" and the return key, the help information of the current statement is displayed. When prefixing

a statement name of the current menu by "?", for example ?LINE, help information of this statement is obtained and the statement becomes the current statement.

Similar device with another special character can be used, if the state of data of the current statement is desired. "=" and the return key is answered by displaying the pairs of the datum names and its values. "=" followed by the datum name, for example =COLOR, is the selective facility of the command. Examining the state of the data of other statements is possible too: all defined data values must be copied to the communication area /MIGVAL/, the statement must be defined as current and the "=" device is ready to use. Help or current state of data facility can be switched off simply redefining the special character to blank.

PRACTICAL EXAMPLE, HOW TO USE THE SYSTEM

Description of the problem

A simple graphical interpreter is chosen to be an overall example of the system. Graphical execution modules are supposed to be prepared already and can do the following:

- Define new rectangular picture: CALL OPEN (dx,dy)
- Define user coordinate system: CALL SYSTEM (xmin,xmax,ymin,ymax)
- Move pen up to x,y: CALL MOVE (x,y)
- Move pen down to x,y: CALL DRAW (x,y)
- Plot the string of characters: CALL TEXT (x,y,fi,h,nch,string)

Language definition

Interpreter language is defined by the statements:

1. PICTURE SIZE dx,dy
2. SYSTEM XMIN xmin XMAX xmax YMIN ymin YMAX ymax
3. MOVE TO x,y
4. DRAW TO x,y
5. TEXT FROM x,y ANGLE fi HEIGHT h STRING "String of characters"

MIGS solution

Batch mode source data for the program MIGDF generating compiled information can look like:

```
TYPE; REAL 1;  VECTOR 2 N=2 ETYPE=1! Vector of 2 elem. of real
      LITERAL 3 LIMIT=40! Literal using ", limited to 40 char.
STATEMENT
DEFINE 1 P:ICTURE (S:IZE,2)
       2 S:YSTEM (XMIN,1 XMAX,1 YMIN,1 YMAX,1)
       3 M:OVE (TO,2);   4 D:RAW (TO,2)
       5 T:EXT (F:ROM,2 A:NGLE,1 H:EIGHT,1 S:TRING,3)
       6 E:XIT;  7 MENU
MENU
DEFINE 1 NAME 7 LIST (1,2,3,4,5,6);  EXIT
```

The datum names and types are defined in pairs. Semicolon determines the minimal recognition length of the names. Single menu with the sequence number 1 named MENU is defined.

The task of the application programmer is to provide the main program and the statement processor module:

```
      PROGRAM  SGRAPH  ! Simple graphical interpreter
      CALL  DEFINE     ! Define working common area
      CALL  MIGRD (2)  ! Read compiled input from unit 2
      CALL  MIGPR (1)  ! Start input displaying menu 1
      END

      SUBROUTINE  MIGEX (ist)  ! Process statement ist
      COMMON  /MIGVAL/  length, v(24)
      GO TO (1,2,3,4,5,6), ist
    1   CALL OPEN (v(1),v(2))
        RETURN
    2   CALL  SYSTEM (v(1),v(2),v(3),v(4))
        RETURN
    3   CALL  MOVE (v(1),v(2))
        RETURN
    4   CALL  DRAW (v(1),v(2))
        RETURN
    5   nch = NCHST (v(5)) ! Determine actual string size
        CALL  TEXT (v(1),v(2), v(3), v(4), nch, v(5))
        RETURN
    6   STOP
      END
```

To draw the rectangle of the size 30 x 10 with the string SOFTWARE TOOL in the middle, the following data must be prepared.

```
PICT 30,10; DRAW 30,0; 30,10; 0,10; 0,0; TEXT FROM 5,4 _
ANGLE 0 H=2  "SOFTWARE TOOL"; EXIT
```

CONCLUSION

The main aim of this work is to show again that the software tool development is the right way to improve the speed and the quality of the technical software production.

REFERENCES

Katzan, H. (1970) Advanced Programming.
Van Nostrand Reinhold,. 68-85

RIB Rechen-und Entwicklungsinstitut für EDV im Bauwesen (1977)
CAD-Bericht CADINT

Wirth, N. (1976) Algorithms + Data Structures = Programs
Prentice-Hall

THE CRITERIA FOR SELECTING AN APPLICATION PACKAGE

Dennis A. Plunkett

General Manager MSS Computer & Business Consultancy Ltd.,
54 Chapel Road, Worthing, Sussex.

INTRODUCTION

If the traditional approach of designing your own 'bespoke' software is to be questioned, the first approach is to examine the Application Packages that are on offer from various sources.

Most of the Mainframe Manufacturers offer to provide, generally under some form of rental, a form of package solution. Often their policy is "change your company working methods to suit our package". This statement is never so clearly stated but their aim is generally in this direction. Naturally so because they cannot afford to support variations to meet specific needs of individual companies. In general terms their packages have been designed to meet most requirements of their largest users and are well designed to integrate with other packages on offer to run on the same operating system.

When we examine application packages in the mini and micro market place, the hardware manufacturer may have a different policy. His interest in the use to which you put his hardware is less positive. Reduced to basics he wants to "sell tin boxes and run like hell". He will often direct you to a third party who has produced some packaged software which will work under the limitations of his system. If this Software House is geographically near you it has advantages for communications and support visits when necessary.

Where do we start?
Good advice is "Buyer Beware". The quality of the software and the level of support you will get are a matter of judgement for you to decide. It is more important to check their expertise in the BUSINESS aspects of their package solution than their knowledge of the hardware and its operating system.

They are both important but WHAT is done is really more important than HOW it does it.

I feel that the second best advice to give anyone is "Buyer Beware - but not too much!" We recently heard of a person who was examining 128 packages to decide which one was best for his company! Perhaps we will hear his results in a couple of years time when he is an 'expert'on the subject.

Bespoke Programming

The statements that an application package

will save time
will be cheaper to obtain
will (presumably) interface with the hardware and the operating system

are perfectly correct BUT - when you define a BESPOKE solution you

define	what you want
design	what you want
program	what you want
implement	what you want

This implies that the processes have only your stated objectives to achieve and no other. It means that the quality of the person doing the systems analysis work has only to understand the 'WANTS' of the user. It is therefore possible for the analyst to have only a fair knowledge of the business area to be systemised. If he is not experienced enough to get a clear definition of requirements from the user, he will provide a solution that is greeted with that famous phrase ... "wouldn't it be an idea if..." This implies either phase 2 or a "slight modification" and all that that entails.

Examining the Package options

Whenever you examine any application package you define what you want

- some of it there - O.K.
- some of it is NOT there - ?
- some of it is more than you want - ?

We have now got greater problems for the Business Systems Analyst to cope with.

- ? Should we change the Company to suite the package?
- ? How to get the things you want that are not there?

- ? Alternatively, do we really want them?
- ? Why don't we want the "goodies" that the package will do? If other companies wanted them, why don't we?
- ? If we DON'T use them NOW, will it be possible to "pick them up" later on without any problem?
- ? If I don't take them, how will we manage if an "enhanced" version of the package expects such data to be available in the future?
- ? How do I get things I MUST have that are not there?

There are also other problems of quality of "services" that will be provided to be evaluated from one package to another. It comes down to the ability of the person being used as the Business Systems Analyst (a title I used deliberately in an earlier paragraph without any other reference).

The case for a Business Systems Analyst

The quality of person to do a BSA job has got to know the business now and in the future. It has also got to be his policy to use the power of the computer to aid the user, not "that will be a problem on the computer - let the user do it". I would describe this as NON User Friendly.

Many other people will describe their system as User Friendly if it only aids any form of input. To be truly USER FRIENDLY the whole aim in the design should be to use the power of the computer to make any work the user has to perform as simple and easy as possible. Minimal input for maximum benefit.

Exploiting the use of the computer has to be in the forefront of the BSA's mind and he should NOT fall into the traditional trap of allowing the users to be "blinded with Science" by the computer staff using DP terminology to avoid doing what the user wants.

Therefore, although it is correct to say a package WILL save time and cost, it will only be a true saving if the BSA's job is done well and there are no hidden costs that only come to light after the package is implemented and these costs are then incurred as a 'fait accompli'.

Implementation phases

In the phases of implementing any system on a computer, including an application package, the special phrases DP people use can be simplified into terms that we have had around in production and engineering for a very long time -

Their relationship to traditional data processing terminology is shown in Figure 1.

D.P. Terminology	General Classification
Project Definition Feasibility Study Background Analysis Prototype Design	Research
Hardware Configuration Systems Flowcharting Systems Specification Programming Implementation Planning Training of Users	Development
Going Live User Acceptance System Review	Production

Figure 1

Defining the Project Management

As anyone will realise, the costs and time for research can be monitored and contained because fewer people are involved so they are easier to control. Out of aborted research work there is generally some benefit even if it is only "learning". It can be repeated if necessary without too much effort and problem and if necessary should be repeated to produce the final "Business Specification".

Development is where money and time get spent and wasted. In-house computer development costs can be 'hidden' where work has been done more than once because the original specification was wrong. Most programs appear to be 80% completed if you enquire - but this does not suggest 80% of time or cost - it is 'opinion'. To complete the remaining 20% always takes a very long time; a case of optimistic rather than pessimistic estimating on the part of the systems staff.

Implementation costs are often not monitored correctly and therefore the cost of a failed system cannot be evaluated. Hence a continual 'system support team' to sort out the problems. They are sometimes called "bugs".

It is on very rare occasions that a hardware fault creates "bugs" in a program. They do not develop because the program has run 'X' times; generally they exist because there were insufficient checks performed at the approval stages to ensure that the program would meet all compound conditions of the business and computer specifications.

Application Package Selection

A suite of programs sold by a reputable software vendor as an Application Package should have been tested in all aspects before being sold. Therefore, you should gain a form of protection from "bugs".

The tried and tested Application Package should be reliable because of this testing in the hands of other users who do not have the expertise of DP staff on hand.

The best Salesperson for any Application Package is a satisfied user. If it is possible to contact one and get his honest opinion without any Salesperson involvement this is the best form of recommendation or demonstration you can obtain. How you achieve it I leave to you to decide.

Software support contracts with the software vendor are a form of insurance for the installation without DP staff on site. We are getting around more and more to the installation where the computer is switched on and off just like any other piece of office machinery by people who do not need to know (apart from back-up routines) any more about 'pure DP' than their basic tasks. A specialist software support Contract is very necessary therefore, if only to cope with legislative changes.

It is up to senior management however to select wisely when they buy an Application Package. Do not leave it to your six months experienced programmer. He will be told by a glib salesperson that "this is the best solution because it's on X technology hardware". Perhaps neither of them really knows the true business application in your company.

An Ideal Situation

The perfect approach, if it were possible to achieve, would I think, come from the following scenario.

A department has a boss due to retire in X months time. His second-in-command, who will replace him, should be nominated 'Business Systems Analyst' and freed from his normal work to develop a new system.

The 'BSA' investigates the hardware and software packages available with reference to DP specialist advice as required. He should select and evaluate and put forward Cost v Benefits proposals for (say) three of the options he puts at the top of his selection list. He then SELLS these ideas to senior management and a final selection is decided and an Implementation Plan WITH COSTS is agreed and BENEFITS are identified.

The BSA should now become the Project Manager to install the plan and hopefully when the "acceptance by user" phase is reached it is by himself as the new head of the department

that the Implementation project is terminated as the Production phase takes over. He is responsible for both phases. Hopefully this coincides with the departure of the previous boss. The ideal is to make your BSA an in-house consultant who has a vested interest in ensuring the successful implementation of a new system.

Who can I turn to?

The role of an external Data Processing Consultant to assist should also be considered, provided his advice is going to be impartial. The sizing of the computer is an area where Consultancy advice is vital to avoid problems. Do not expect the hardware salesman to give it - he has one objective - a sales target to achieve to give him his commission.

Consultancy advice may (in UK) be supportable by a Government Grant in certain circumstances. Approval must always be obtained first; it is never approved in retrospect and generally only covers the 'Feasibility Study'.

As a consultant, the main reason I hear from clients for developing bespoke programs is "because we are different to anyone else". Dare I say that what we, as outsiders, quite often see is how similar one company is to another even if their products are different.

I think senior management have for a long time abdicated to DP management regarding new systems because of senior management's fear of getting to grips with 'Computers'. User management should realise that if they choose an application package that they understand, how it does what they want, irrespective of too much involvement with how the computer does it, we will get better business applications working faster and not worry too much about complexities of "integrated data bases, distributed networks, on line, Real Time, etc. etc. etc."

Areas covered by application packages have now passed well beyond the basic accounting systems. There are packages available on mainframes, minis and micros covering most industries. Some of the ideas have been sponsored by trade associations developing industry standard packages.

The office of the future is a good concept but we have got to quickly become more efficient now so that we survive to plan such ideas.

HYDRAULICS

PROGRAM PACKAGE USED FOR OPTIMAL CONTROL OF THE DJERDAP HYDROPOWER SYSTEM

Milorad Miloradov* Serafim Opricovic+ Branislav Djordjevic+

*"Jaroslav Cerni" Institute for Development of Water Resources
+ Faculty of Civil Engineering, University of Belgrade,
Yugoslavia.

SUMMARY

The two national power systems (Yugoslav and Rumanian) share the hydropower of the Djerdap (The Iron Gate) system on the river Danube. The optimization model should enable the managers to operate optimally this hydropower system of two large reservoirs with two hydropower plants at each of two dams. The model is realized as a program package with appropriate data base. The program package consists of an interactive main program, few basic programs for the solution of optimization tasks, and several subroutines for particular computations. The interactive mode enables the user to activate the desired procedure for the program package.

1. SYSTEM DESCRIPTION

On the border between Yugoslavia and Rumania, one of the biggest hydropower plants - Djerdap I - was built on the river Danube. The total discharge capacity is 9,600 m^3/s and the net head ranges between 21.9m and 34.5m. The plant can produce 2,300 MW, while the average annual production is around 12,000 GWh. The Kaplan turbines with power of 190 MW and a runner whose diameter equals 9.5m are the biggest turbines of this type /1/.

In 1979, the construction of a new hydropower plant was begun downstream of the already existing one. It was named the Djerdap II. This new plant will consist of two parts (the Yugoslav and the Rumanian one), equipped with 2 x 8 turbular turbines of 27.5 MW each and a runner diameter of 7.5m. The mean head of this plant will be 7.45m, while the minimum head which will enable the stable operation of the power plant will be 2.5m. The total power of the plant will be 440 MW, and an annual energy production of around 2,600 GWh is anticipated.

It is typical for this system that the backwater was realized under conditions of a lowland alluvial river, with relatively low banks. Because of this, it is essential to provide complex protection of the lowlying land along the banks by means of levees and drainage systems. Therefore, there are constraints on the water level at some characteristic points along the backwater, which is around 300km long.

Upstream of the Djerdap I three big tributaries flow into the Danube: the Tisa, the Sava and the Velika Morava. They are also affected by the backwater. The part of the Danube between hydropower plants Djerdap I and Djerdap II have no big tributaries.

There also are no big tributaries at the downstream part of the Danube, up to the Yugoslav - Bulgarian border, so that the flow of this part could be modelled as the flow between the plants. For this part of the Danube there are no constraints on the water levels but only on the daily oscillations of the level under the conditions of peak operation in regard to the medium daily level. The constraints on minimum flow are caused by the navigation conditions.

The entire system consists of threee sub-systems:

- Djerdap I storage with tributaries and power plant,
- Djerdap II storage and hydropower plant,
- watercourse downstream from the dam Djerdap II.

The outflow of the Djerdap I is input into the Djerdap II, and the output of the Djerdap II is input into the downstream part of the watercourse. All three parts represent one entity and influence the optimal operation of each hydropower plant.

The construction of the downstream hydropower plant - the Djerdap II - raises new optimization and control problems. Both the Yugoslav and Rumanian power systems now demand more and more of peak power and energy. By including the downstream reservoir into the system, it is possible to produce more peak power with the upstream power plant, ie. to realize a considerably greater production of energy during critical intervals of a day or week. It is quite obvious that this regime of operation means unsteady flow in both reservoirs. Therfore, the optimization model of the system should include the mathematical model of unsteady flow /2/ in both reservoirs and downstream of them.

2. THE AIM OF THE OPTIMIZATION MODEL

The optimization model should enable the dispatchers of the Djerdap to control optimally this complex system. The main aims of the optimal operation are:

- utilization of the available hydropower potential,
- optimal energy production (providing maximum peak energy),
- production of energy required by the two national power systems,
- optimal utilization of the units of power plants,
- minimum spillover flows.

By analizing the aims it could be seen that the problem of optimal exploitation of the Djerdap I and II could be stated in one of the following forms:

a) maximization of the total produced energy

$$E_{max}(T) = \max_{\{Q^I,\, Q^{II}\}} \int_0^T \left[P^I(t) + P^{II}(t)\right] \, . \, dt$$

where:

T = time horizon (seven days or shorter)
Q = discharge through the power plants
P = power
I and II = indicators for the Djerdap I and Djerdap II,

b) maximization of the production benefit

$$D_{max}(T) = \max_{\{Q^I,Q^{II}\}} \int_0^T D\,(P^I(t),\, P^{II}(t)) \, . \, dt$$

where:

D = benefit of producing power, considering the different price of peak and basic power,

c) production according to the diagram of required power, $P^R(t)$ (of energy) with optimal distribution of power between two plants.

$$\min_{\{Q^I,Q^{II}\}} \int_0^T \left[P^R(t) - P^I(t) - P^{II}(t)\right]^2 \, . \, dt$$

The solution of these optimization problems must satisfy all the constraints on the unsteady flow, and on the water levels in the reservoirs /3/. The constraints are defined by the Convention and the Agreement on Hydropower plants utilization.

From the hydropower plant point of view it would be most convenient to optimize the energy production according to maximum benefit. However, the question of fitting such production in the power systems of the two countries is raised. That is why the task of the optimal fitting of the possible production in the requested one is also imposed.

In addition to the solution of the stated problems, the optimization model should satisfy the demands which could occur during the use of the model such as:

- production according to temporary control strategy
- operation of only one power plant
- change or replacing one of the constraints
- change of the required power at any moment.

The model should enable the correction of the production plan if a considerable difference occur between the computed (planned) values and the realized ones.

Besides the production planning, the model should register the power production and the states in the reservoirs in real time. It should calculate the difference between planned and produced energy, and the balance of energy distribution to the users (national power systems).

Taking all these into account, it is obvious that the model has to be of the modular type, composed from few programs which will be connected in one logical entity with the possibility of the interactive execution. All the constraints that could be changeable have to be treated parametrically so that they could be changed easily.

3. PROGRAM PACKAGE

The optimization model is realized as a program package for the computer with appropriate data base, Fig 1. The model consists of the man-model interactive part, few basic algorithms (or programs) for the solution of the tasks mentioned in previous chapter, and several sub-programs for particular computations. The structure or the program package results from the desired model structure.

The user of the program package enter the data into the data base directly, or by the appropriate routines. The results from the data base are obtained by the output routines. The connection with the data base could be established by the interactive part of the model, too. The main function of the interactive part is to enable the user to activate the desired computing procedure, ie. program, for the program package. The data from the real system and realized values of all variables

encompassed by the model are obtained by measuring instruments and will be stored in the data base automatically with the fixed period of sampling.

The interactive part of the model is presented in Fig 2. When the program package is activated, the question "SYSTEM?" appears on the display, with a remark that there are two possible answers: 1) SYST and 2) DJEI. The integer 1 is entered when the user wants to use the model for the entire system, and 2 only for the Djerdap I power plant and the upstream river flows. The second question which appears is "PROGRAM?", after which a list of names of main programs from the package appears. One program number (NP) from the list is entered as the answer.

The maximum time horizon of the optimization model is seven days. The start and duration (t_o,T) of the time horizon are given after the question "TIME?" entering the starting hour and day, and the end day and hour in the week for the needed computations.

By entering the program parameters values after "PARAMETERS?", the user of the program package may determine the way of the program usage. For example, by the parameters, the set of conditions and constraints, that have to be respected in the optimization computations, is given.

The user can activate one of the following programs from the program package:

1. MAXIENER - computations of production effect with the maximum (long-term) amount of energy,
2. OPTIENER - computations for the operation with optimal benefit in the system,
3. ENDEMAND - computations for the production according to the given diagram of energy demand,
4. OPTIMAST - application of dynamic programming for determination of optimal control strategies,
5. GIVCONTR - computations for the production according to the given control (or trajectory) at a control point,
6. TESTCONT - testing of the difference between the realized production and the planned one,
7. BALANCEN - computation of the produced peak and basic energy for the Yugoslav and Rumanian power systems, the difference from the planned production, and rebalance for the next time period.

The programs mentioned call sub-programs for particular analysis and computations. More important sub-programs are: UNSTEFLOWI for the computation of unsteady flow in the upper reservoir with the tributaries, UNSTEFLOWII for the compuatation of unsteady flow in down reservoir and downstream, ENERGYVL for the comp-

utations of the power or energy values with the known values of the levels and discharge, NUMTURBO for determining the necessary number of the units to produce the given value of energy.

The programs receive all input data from the data base. The results are stored into the data base, but only particular messages are immediately transfered (displayed) to the package user. By the output routines the user could get the results in the desired form.

All the programs from the package use the data base through the particular subroutines - Fig 1. Applying this approach the main programs are independent of the structure of data base. The eventual changes of the files cause only the changes into the input or output subroutines.

4. DATA BASE

The operative data base contains all data used by the program package for the required computations. The original input data are entered by user of the program package according to the specified formats. Corrections and updating could be performed by support of the computer software (Editmode). Some of the input data are transformed into the computer code and read by the "Read" subroutines - Fig 1. The inter-results, computed by one of the programs and used as input data by others, are "written" and "read" without formats (stored in the computer code).

The data base consists of several files. Some of them contain: morphological data (of the river system), characteristics of the units of power plants, spillways and navigation locks, and extreme values of controlable variables. Most of the files contain the discrete values of the dynamic variables during the given time horizon. The model will deal with the average hourly values, so the record contains 7 x 24 (maximum 7 days) values of one variable. By the particular subroutine the record (block) is used as a ring which contains the data for seven days in a week. The files for dynamic variables contain: required power, predicted hydrological input (inflows), and planned values of power, water levels and flows (program results).

The data from the real system (water levels), and realized values (powers) will be obtained by measuring instruments and stored into data base. The file with the measuring values has similar structure as the file with planned values, but it contains the data for the past seven days. Measuring is defined by the particular project of the information system of the Djerdap.

To keep the old data the contents of file could be transfered into the archival data base (on tape) with corresponding date.

The contents of the operative data base should be transfered into the archival at the fixed hour and day in the week.

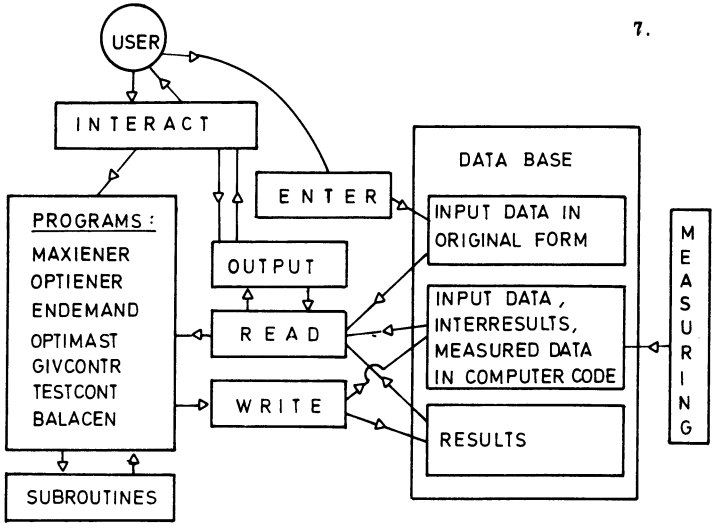


Fig. 1- SCHEME OF PROGRAM PACKAGE AND DATA BASE

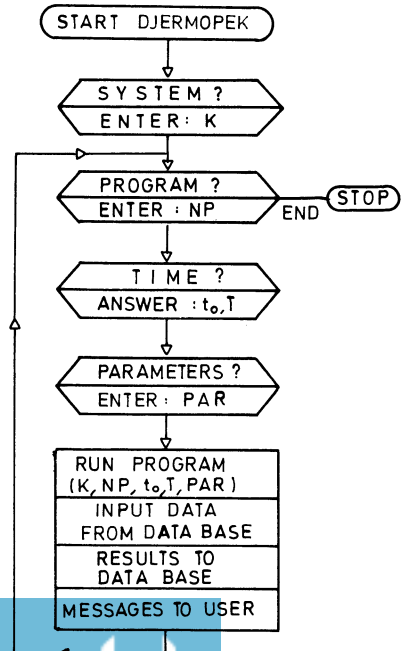


Fig. 2 - RUNNING PROGRAMS BY INTERACT

ACKNOWLEDGEMENT

The research reported herein is supported by the Serbian Community of Sciences - Beograd, and by the working orgainization HE Djerdap, Kladovo.

REFERENCES

1. Jakovljević P : Power Plants in the Yugoslav - Rumanian Reach of the Danube (in Serbo-Croat), Niro Export Press, Beograd, 1979.

2. Moloradov M . Opricović S and P Tanasković : "The Mathematical Model of Unsteady Flow and Its Application to Water Reservoir Control with Daily Regulation", Proc. 2nd International Conference on Applied Numerical Modelling, Madrid, Sept. 1978.

3. Preliminary project "The Mathematical Model of Optimal Exploitation of the Hydropower Plants Djerdap I and II" (in Serbo-Croat), "Jaroslav Cerni" Institute, Beograd, Dec. 1981

DEVELOPMENT OF A SOLUTE TRANSPORT MODEL FOR MULTI-LAYER GROUNDWATER BASIN

A. Das Gupta and W.B.R. De Mel

Asian Institute of Technology, P.O. Box 2754
Bangkok, Thailand

SUMMARY

A quasi-three-dimensional finite difference solute transport model has been developed with a view to apply in solving a field problem of saltwater encroachment in a multi-layer groundwater basin. The flow model is coupled with the mass transport model for multi-layer system to simulate the changes of solute concentration in all aquifers based on the piezometric head changes. The interaction between the aquifers is included through leakage flux calculated on the basis of the piezometric head gradient existing across the intervening aquitards. The model has the flexibility of using non-uniform grid spacing and time step to facilitate its efficient application for an extensive regional basin. The extent of applicability of the model to the actual field situation is checked by solving three hypothetical problems involving two confined aquifers.

INTRODUCTION

Many groundwater basins, after years of development, face the problem of salt contamination in the producing areas, if proper management on the utilisation is not exercised. The migration of saltwater into freshwater formations is mainly due to the reversal or reduction of the piezometric head gradient, which allows heavier saline water to move into the regions of permeable formation where freshwater resides. The contaminating source may be one or a combination of the following: (1) sea water in coastal areas; (2) saline water that was trapped in sediments during deposition; and (3) that invaded the sediments during periods of high sea elevation in past geologic times. To alleviate this problem, the first thing is to understand the mechanism of contamination, then to identify the contaminating source and subsequently to be able to predict the spatial and temporal variation of salinity under different feasible control measures. The problem of solute movement through saturated for-

mations has been studied both analytically and numerically by many researchers. But few reported investigations by Pinder (1973) of contaminant distribution in an aquifer of glacial sediments; by Bredehoeft and Pinder (1973) of solute transport in limestone aquifer; by Schwartz (1977) in a very low permeability sediments of shales, silts and glacial till; and by Das Gupta and Yapa (1982) in an unconsolidated aquifer of sand and gravel are related to field problems on a regional scale.

In this context, the present paper deals with the development of a quasi-three-dimensional finite difference solute transport model for a multi-layer groundwater basin with the objective of obtaining a practical model that combines the virtue of a sound theoretical basis and simple mathematical manipulation and computer programming and closely represents the hydrogeological situations in a specific problem area in Thailand. The spatial and temporal variations of salinity in water bearing formations are simulated considering convection and dispersion transport phenomena in response to the hydrodynamic behaviour of the multi-layer system simulated by a quasi-three-dimensional flow model. The interaction between water bearing layers is included through leakage flux calculated on the basis of the piezometric head gradient existing across the intervening layer. The iterative alternate direction implicit procedure is used to solve the resultant finite difference equations. The computer code developed for the model consists of a special technique to reduce the storage requirement, so that a multi-layer system with several layers can be solved using a relatively small storage. The extent of applicability of the model to the actual field problem has been checked by solving three idealised systems involving two confined aquifers, pumped at the centre of the area. The idealised systems considered are: (1) a connate water body is entrapped in the first aquifer; (2) saltwater seeps from a river into the upper aquifer; and (3) both the aquifers are open to the sea along one boundary. A complete elaboration of the model with its application as mentioned is provided in the following sections.

MATHEMATICAL MODEL

The physical - chemical processes involved in solute transport are complex in nature. There are three distinct processes that contribute to transport: convection, mechanical dispersion and molecular diffusion. A general discussion on transport processes has been given by Bower (1978) and Freeze and Cherry (1979). In solute transport in aquifers the contribution of diffusion is usually negligible in comparison to that of mechanical dispersion provided that the transport is not in unfractured clays, clayey silts and shales where the groundwater velocities are normally low and where diffusion can be significant (Pickens and Lennox, 1976). The model consists of two parts, the first one deals with groundwater flow aspects and

the predicted piezometric head distribution in aquifers are then utilised in the second model to simulate the saltwater movement.

Basic concept

The anticipated hydrodynamic behaviour of a multi-layer system is that with an initial piezometric head distribution in the system as static, if water is withdrawn from any aquifer the piezometric head drop in that aquifer occurs first locally and later spreads to other areas in the aquifer as pumping is continued. This lowering of the piezometric head in that particular aquifer disturbs the equilibrium of the system. This initial disturbance will be first absorbed by adjacent aquitards releasing water from their storages, but when the disturbance reaches the far ends of the aquitards the adjacent two aquifers are affected. During this stage, water will flow to the aquifer being pumped from aquitard storage and from the adjacent two aquifers. In the third stage, water released from the aquitard storage is negligible and the flow will be due to direct leakage from the adjacent two aquifers through the aquitards. In principle, the leakage flux during all three stages should be included. However, when the leakage fluxes during the first two stages are included, the model becomes more complicated mathematically. Conceptually, the transport of solute in the aquifer is governed by convection and dispersion process. Also, in a multi-layer system, it is expected to have transport of solute from one aquifer to another through the intervening medium depending on the magnitude and direction of vertical flux and intervening medium characteristics. Mathematically this aspect can be incorporated in the modelling procedure, but non-availability of characteristic parameters to quantify the dispersive behaviour of the intervening medium in practical situations warrants its applicability. Considering the views expressed above, the following simplifications are made:

1. The aquitard storage is ignored when the leakage flux is estimated, but included when the piezometric head change in the aquitard is subsequently required. The computational effort required to estimate the piezometric head changes in the aquifers is much less compared with that needed when the effect of aquitard storage on leakage is incorporated. The simplification is justified since the piezometric head in the aquifers is not significantly affected, and that only over a time scale shorter than of practical interest. Moreover, the more complicated mathematical procedure alone may not improve the ultimate solution, since model parameters are only approximate in most practical situations.
2. It has often been observed that the piezometric head in an aquifer does not vary significantly with thickness and as such the aquifer is represented as a planar surface between the two aquitards.

3. In general the piezometric head variation in aquitards in the vertical direction is greater than that in the lateral direction and therefore the flow in aquitards is assumed to be vertical.
4. The water quality problem to be solved in the practical situation is of chloride contamination as such energy transport and chemical reactions are neglected. The concentrations observed in the aquifers under consideration were sufficiently low so that a constant density fluid could be assumed. The groundwater flow pattern is not altered by the presence of the salt in solution.
5. The transport of solute between aquifers through the intervening aquitards is mainly considered as convection dominated.

The physical configuration of an idealised multi-layer system is shown in Figure 1a. An aquitard and aquifer immediately below it is considered as one stack; stacks are numbered from top to bottom. The simplified system for modelling purpose is shown in Figure 1b and the model representation of the physical system at any specific point is in Figure 1c. Cartesian axes x and y define the horizontal plane and coordinate axes z is taken to be vertically downward.

Flow equation

The partial differential equation governing the piezometric head variation in aquifer ℓ is (Pinder and Bredehoeft, 1968)

$$\frac{\partial}{\partial x}(T_x \frac{\partial h}{\partial x})_\ell + \frac{\partial}{\partial y}(T_y \frac{\partial h}{\partial y})_\ell + W = S_\ell \frac{\partial h_\ell}{\partial t} \qquad (1)$$

where T_x and T_y are transmissivities of the aquifer in x and y directions, respectively; S represents the storativity of the aquifer; h stands for the piezometric head in the aquifer; W represents the sum of flows going out and coming into the aquifer, expressed as a flux; and subscript ℓ indicates that all parameters are referring to the particular aquifer under consideration.

The term W includes discharge (Q), recharge (R) and the leakage flux through aquitards from aquifers lying above and below the aquifer under consideration and is expressed as

$$W = - Q + R + q_\ell - q_{\ell+1} \qquad (2)$$

The two leakage flux terms q_ℓ and $q_{\ell+1}$ are proportional to the piezometric head gradient in the aquitards at the boundary surfaces in contact with aquifer ℓ

$$q_\ell = - K'_\ell (\frac{\partial H_\ell}{\partial z})^L \qquad (3)$$

$$q_{\ell+1} = - K'_{\ell+1} (\frac{\partial H_{\ell+1}}{\partial z})^U \qquad (4)$$

where K'_{ℓ} and $K'_{\ell+1}$ are the vertical hydraulic conductivities of aquitards ℓ and $(\ell+1)$ respectively; H_{ℓ} and $H_{\ell+1}$ stand for the piezometric head in aquitards ℓ and $(\ell+1)$ respectively; and superscripts L and U denote the particular quantities at the lower and upper boundaries of the corresponding aquitard. The piezometric head variation in the aquitard ℓ is given by

$$\frac{\partial^2 H_{\ell}}{\partial z^2} = \frac{S'_{\ell}}{K'_{\ell}} \frac{\partial H_{\ell}}{\partial t} \tag{5}$$

where S'_{ℓ} and K'_{ℓ} denote the storage factor and vertical hydraulic conductivity of the aquitard ℓ respectively.
In addition, the piezometric heads at the upper and lower boundaries of the aquitard ℓ should be equal to the piezometric head in aquifers overlying and underlying the aquitard.

Mass transport equation

The spatial and temporal variation of solute in an aquifer due to convection and dispersion in a multi-layer system can be expressed as (Bredehoeft and Pinder, 1973)

$$\frac{\partial}{\partial x}\left(D_x \frac{\partial c}{\partial x}\right)_{\ell} + \frac{\partial}{\partial y}\left(D_y \frac{\partial c}{\partial y}\right)_{\ell} - \frac{\partial}{\partial x}(uc)_{\ell} - \frac{\partial}{\partial y}(vc)_{\ell}$$

$$= \left(\frac{\partial c}{\partial t}\right)_{\ell} + \left(\frac{cQ}{b\theta}\right)_{\ell} + \frac{q_{\ell}}{b_{\ell}\theta'_{\ell}} + \frac{q'_{\ell}}{b_{\ell}\theta'_{\ell}}$$

where c is the concentration of solute in the aquifer (mass of solute per unit volume); D_x and D_y are coefficients of dispersion in x and y directions respectively; u and v represent the seepage velocities in the aquifer in x and y directions respectively; Q stands for the withdrawal rate as a flux from the aquifer; q is the direct leakage flux between the aquifers $(\ell-1)$ and ℓ; q' is the direct leakage flux between the aquifers ℓ and $(\ell+1)$; b stands for the thickness of the aquifer; and θ' represents the porosity of the aquitard. The direct leakage terms are expressed as

$$q_{\ell} = - c_{\ell-1} \frac{K'_{\ell}}{b'_{\ell}} \delta(h_{\ell-1} - h_{\ell}) + c_{\ell} \frac{K'_{\ell}}{b'_{\ell}} \delta(h_{\ell} - h_{\ell-1}) \tag{7}$$

$$q'_{\ell} = c_{\ell} \frac{K'_{\ell+1}}{b'_{\ell+1}} \delta(h_{\ell} - h_{\ell+1}) - c_{\ell+1} \frac{K'_{\ell+1}}{b'_{\ell+1}} \delta(h_{\ell+1} - h_{\ell}) \tag{8}$$

where b'_{ℓ} and $b'_{\ell+1}$ are the thickness of aquitards ℓ and $(\ell+1)$ respectively; h_{ℓ}, $h_{\ell-1}$ and $h_{\ell+1}$ are the piezometric heads in aquifers ℓ, $(\ell-1)$ and $(\ell+1)$; and δ is the dirac delta function. The coefficients of dispersion are related to the dispersivity of the medium and the magnitude of the resultant seepage velocity as

$$D_x = \alpha_x \bar{V} \tag{9}$$

$$D_y = \alpha_y \bar{V} \tag{10}$$

where α_x and α_y are dispersivities of the medium in x and y directions respectively. The components of the resultant seepage velocity in x and y directions are

$$u = -\frac{K_x}{\theta}\frac{\partial h}{\partial x} \tag{11}$$

$$v = -\frac{K_y}{\theta}\frac{\partial h}{\partial y} \tag{12}$$

where K_x and K_y are hydraulic conductivities of the aquifer in x and y directions respectively; and θ is the porosity of the aquifer.

NUMERICAL FORMULATION

For numerical approximation of the two governing equations, the aquifer area is discretised into several cells by two sets of lines parallel to the x and y axis as shown in Figure 2a. Spacing between lines in each set may not be uniform so that usually the cells are rectangles of different sizes. The aquifer properties in each cell are assumed constant and are represented at the central point of the cell known as node. The variation of parameters over the aquifer at any time are specified by nodal values. Also the variations of parameters with time at any node are specified at different time intervals which need not be spaced uniformly. A general node point (i,j) and its four surrounding nodal points as shown in Figure 2a are considered for a particular aquifer for developing the finite difference form of the governing equations. Also the leakage fluxes at node (i,j) from the adjacent aquifers through the intervening aquitards, as shown schematically in Figure 2b, are assumed to be proportional to the piezometric head gradient across the aquitard and the solute transport in vertical direction is due to convection mechanism only.

Flow equation

Considering the above two components of regional flow in an aquifer and the vertical leakage flux between aquifers, the finite difference form of Equation 1 is expressed as

$$A\, h^n_{i-1j\ell} + B\, h^n_{i+1j\ell} + C\, h^n_{ij-1\ell} + D\, h^n_{ij+1\ell} + E\, h^n_{ij\ell}$$

$$= W'_{ij} - P\, h^{n-1}_{ij\ell} - \left(\frac{K'}{b'}\right)_{ij\ell}\left(h^n_{ij\ell-1} - h^n_{ij\ell}\right) + \left(\frac{K'}{b'}\right)_{ij\ell+1}$$

$$\left(h^n_{ij\ell} - h^n_{ij\ell+1}\right) \tag{13}$$

where, the third subscript ℓ denotes the aquifer and aquitard

number in the system; superscript n denotes the time interval;

$$A = T_{x_{i-\frac{1}{2}j}} / (\Delta x_{i-\frac{1}{2}} \Delta x_i); \quad B = T_{x_{i+\frac{1}{2}j}} / (\Delta x_{i+\frac{1}{2}} \Delta x_i);$$

$$C = T_{y_{ij-\frac{1}{2}}} / (\Delta y_{j-\frac{1}{2}} \Delta y_j); \quad D = T_{y_{ij+\frac{1}{2}}} / (\Delta y_{j+\frac{1}{2}} \Delta y_j)$$

$$E = -(A+B+C+D+P); \quad P = S_{ij}/\Delta t; \text{ and}$$

$$W'_{ij} = Q_{ij}/(\Delta x_i \Delta y_j)$$

The finite difference form of Equation 5 governing the piezometric head variation in the aquitard is expressed explicitly as

$$H_k^{n+1} = F_o(H_{k+1}^n + H_{k-1}^n) + (1-2F_o)H_k^n \qquad (14)$$

with $F_o = (K' \Delta t/S' \Delta z^2)$

In Equation (14), H_k^{n+1} and H_k^n represent the piezometric heads at node k of the aquitard at the end of time step (n+1) and n, respectively.

Mass transport equation

The finite difference equivalent of the governing equation, Equation 6, with substitution of direct leakage (Equations 7 and 8) and dispersivity coefficients expressed as in Equations 9 and 10 has been developed with central difference scheme for the space derivatives and forward difference scheme for time derivative

$$A\, c_{i-1j\ell}^n + B\, c_{ij\ell}^n + D\, c_{i+1j\ell}^n + E\, c_{ij-1\ell}^n + F\, c_{ij+1\ell}^n - R\, c_{ij\ell-1}^n - S\, c_{ij\ell+1}^n = -\frac{1}{\Delta t} c_{ij\ell}^{n-1} \qquad (15)$$

where $A = u_{ij}/2\Delta x_i - \frac{\alpha_x}{4\Delta x_i^2}(\bar{V}_{i+1j} - \bar{V}_{i-1j}) + \frac{\alpha_x \bar{V}_{ij}}{\Delta x_i (x_i - x_{i-1})}$;

$$B = -\left\{\frac{\alpha_x \bar{V}_{ij}}{\Delta x_i}\left(\frac{x_{i+1} - x_{i-1}}{(x_{i+1} - x_i)(x_i - x_{i-1})}\right) + \frac{\alpha_y \bar{V}_{ij}}{\Delta y_i}\left(\frac{y_{j+1} - y_{j-1}}{(y_{j+1} - y_j)(y_j - y_{j-1})}\right)\right.$$

$$+ (u_{i+1j} - u_{i-1j})/2\Delta x_i + (v_{ij+1} - v_{ij-1})/2\Delta y_i + Q_{ij}/\theta_\ell b_\ell$$

$$+ \frac{K'_{ij\ell}}{\theta_\ell b_\ell b'_{ij\ell}} \delta(h_{ij\ell} - h_{ij\ell-1}) + \frac{K'_{ij\ell+1}}{\theta_\ell b_\ell b'_{ij\ell+1}} \delta(h_{ij\ell} - h_{ij\ell+1})$$

$$\left. + \frac{1}{\Delta t} \right\} ;$$

$$D = \frac{\alpha_x}{4\Delta x_i^2}(\bar{V}_{i+1j} - \bar{V}_{i-1j}) + \frac{\alpha_x \bar{V}_{ij}}{\Delta x_i (x_{i+1} - x_i)} - u_{ij}/2\Delta x_i \quad ;$$

$$R = \frac{K'_{ij\ell}}{\theta_\ell b_\ell b'_{ij\ell}} \delta(h_{ij\ell-1} - h_{ij\ell}) \quad ;$$

$$S = \frac{K'_{ij\ell+1}}{\theta_\ell b_\ell b'_{ij\ell+1}} \delta(h_{ij\ell+1} - h_{ij\ell}) \quad ; \text{ and coefficients E and F}$$

are expressed similar to coefficients A and D respectively with the indices changed for y-direction.

SOLUTION SCHEME FOR FINITE DIFFERENCE EQUATIONS

A modified form of the alternating direction implicit method of Peaceman and Rachford (1955) has been used to solve the resultant set of equations. The procedure is same for both the equations and the reference is being made only to the mass transport equation. In principle, from the known variation of concentration over aquifers at any time, Equation 15 can be used to find the state of the system at different nodes progressively at discrete time intervals. With this process, for a regional multi-layer system, the number of unknowns to be solved at any instant will be enormous. In the modified alternating direction implicit method, Equation 15 is rearranged in two forms as follows

$$A\, c^n_{i-1j\ell} + B\, c^n_{ij\ell} + D\, c^n_{i+1j\ell} = -\frac{1}{\Delta t} c^{n-1}_{ij\ell} - E\, c^n_{ij-1\ell} - F\, c^n_{ij+1\ell} + R\, c^n_{ij\ell-1} + S\, c^n_{ij\ell+1} \qquad (16)$$

$$E\, c^n_{ij-1\ell} + B\, c^n_{ij\ell} + F\, c^n_{ij+1\ell} = -\frac{1}{\Delta t} c^{n-1}_{ij\ell} - A\, c^n_{i-1j\ell} - D\, c^n_{i+1j\ell} + R\, c^n_{ij\ell-1} + S\, c^n_{ij\ell+1} \qquad (17)$$

The unknown concentration values on the left hand side of Equation 16 correspond to nodal points along one grid line parallel to the x-axis. When this equation is applied along one such grid line with a number of nodes NY, the resulting set of NY simultaneous equations can be solved for the NY unknown nodal concentration values if some values are assigned for the unknown nodal concentration values on the right hand side. Similarly, the concentration values at the nodal points can be obtained by applying Equation 17 along grid lines parallel to the y-axis. These two equations are threrefore alternately used in the scheme to correct the nodal concentration values and the solution method is iterative. The concentration

values are first predicted by the simple formula

$$\frac{c^{n}_{ij\ell} - c^{n-1}_{ij\ell}}{c^{n-1}_{ij\ell} - c^{n-2}_{ij\ell}} = \frac{c^{n-1}_{ij\ell} - c^{n-2}_{ij\ell}}{c^{n-2}_{ij\ell} - c^{n-3}_{ij\ell}} \qquad (18)$$

Then these approximate nodal values are corrected repeatedly by using Equations 16 and 17 until they converge to the desired accuracy. When equation 16 or 17 is applied to any node at any iteration, the most recently estimated values are assigned to the nodal points in the right hand side of the equation. During each iteration, the nodal values in any aquifer are corrected twice consecutively, first along grid lines parallel to the y-axis (Equation 17) and then along grid lines parallel to the x-axis (Equation 16). At the end of each iteration a convergence test is applied to the nodal values. In the present model, the sum of the improvement of all nodal values during the iteration is taken as an indicator and if this is smaller than a pre-assigned tolerance, the iteration is discontinued.

APPLICABILITY OF THE MODEL

The model has been applied to three hypothetical conditions to illustrate its applicability to field problems. An aquifer system of two stacks with a central pumping well as shown in Figure 3 has been considered to simulate the transport of saltwater when the source is defined by the existance of a connate water body in the first aquifer (case a); by seepage from a saltcontaminated river having hydraulic contact with the first aquitard (case b); and by direct hydraulic contact with sea along one boundary of the domain (case c). The plan view of the aquifer stack with the specified condition for the three cases is shown in Figure 4. The input parameters and other relevant data for the model are as follows: hydraulic conductivity of aquifers = 80 m/day; hydraulic conductivity of aquitards = 48 x 10^{-5} m/day; thickness of aquifers and aquitards = 20 m; storativity of aquifers = 1 x 10^{-4}; dispersivity of the medium = 500 m; porosity = 0.3; initial chloride concentration in two aquifers = 0; chloride concentration of the source of contamination = 10,000 $mg\ell^{-1}$; river water level above the reference plane = 102 m; size of the aquifer system = 17x17km; and period of simulation for salt transport = 20 years. Initially the piezometric heads are assumed to be constant over the area, at 100 m for the first aquifer and at 95 m for the second aquifer. After a period of five years of pumping at a rate of 1,000 m^3/day from each aquifer at the centre of the area, the simulated piezometric head distributions for the two aquifers are shown in Figure 5. These distributions are considered constant for the next twenty years to simulate the transport of saltwater under three conditions.

For case (a), with connate water body of 10,000 $mg\ell^{-1}$ concentration in the first aquifer, the saline plume advances radially in the first aquifer and in the second aquifer due to the leakage flux from the first aquifer (Figure 6). Close to the source, the concentration gradient is quite high in the first aquifer, whereas, further away from the source, the rate of encroachment of the saltwater front defined by 10 $mg\ell^{-1}$ contour is nearly same in both the aquifers. The plume is identical about the central radial line which indicates that the transport is correctly simulated by the model. The case (b)has many field applications where the aquifer gets contaminated from seepage water of the contaminated river. The distribution of chloride concentration in two aquifers after twenty years of simulation period is shown in Figure 7. As expected, the contaminated plume is symmetrical about the central axis of the area and the rate of advancement of the plume is faster in the central portion. Close to the source, the first aquifer shows higher concentration, while further away from the source, both the aquifers show more or less same degree of concentration distribution. Intrusion of saltwater from the sea is common in coastal areas. Pumping from aquifers creates an inland piezometric head gradient which causes the saltwater intrusion and this situation is schematically represented as case (c). With both aquifers having hydraulic contact with sea, where salt concentration is 10,000 $mg\ell^{-1}$, the simulated distributions of chloride concentration in two aquifers are shown in Figure 8. The concentration plume is symmetrical about the central axis in both the aquifers. The plume advances more in the second aquifer because of vertical leakage flux from the first aquifer. Further, in both aquifers, the plume has a faster movement in the central portion of the area.

CONCLUSIONS

A finite difference numerical model for solute transport in multi-layer groundwater basin has been presented. The interaction between the aquifers is included through leakage flux calculated on the basis of the piezometric head gradient existing across the intervening aquitards. A quasi-three-dimensional flow model is coupled with a quasi-three-dimensional mass transport model to simulate the transport of solute when the contaminating source can be entrapped connate water bodies; seepage from contaminated streams; and saltwater from the sea. These three conditions to be prevalent in the field situation where the model will be used for identification and prediction purposes. A modified form of alternate direction implicit method has been used for solving the resulting set of finite difference equations. The model has the flexibility of using non-uniform grid spacing and time step to facilitate its efficient application for an extensive regional basin. Three hypothetical problems involving two confined aquifers considered for discussion purpose demonstrate the applicability of the mo-

del to the actual field situation.

REFERENCES

Bower, H. (1978) Groundwater Hydrology. McGraw-Hill, New York, 480.

Bredehoeft, J.D. and Pinder, G.F. (1970) Digital Analysis of Areal Flow in Multiaquifer Groundwater Systems: A Quasi-three-dimensional Model. Water Resources Research, 6, 3: 883-888.

Das Gupta, A. and Yapa, P.N.D.D. (1982) Saltwater Encroachment in an Aquifer: A Case Study. Water Resources Research, 18, 3: 546-556.

Freeze, R.A. and Cherry, J.A. (1979) Groundwater. Prentice-Hall, Englewood Cliffs, New Jersey, 604.

Peaceman, D.W. and Rachford, J.R. (1955) The Numerical Solution of Parabolic and Elliptical Difference Equations Journal of Society of Industrial and Applied Mathematics, 3, 11: 29-41.

Pickens, J.F. and Lennox, W.C. (1976) Numerical Simulation of Waste Movement in Steady Groundwater Flow Systems. Water Resources Research, 12, 2: 171-180.

Pinder, G.F. and Bredehoeft, J.D. (1968) Application of the Digital Computer for Aquifer Evaluation. Water Resources Research, 4, 5: 1069-1093.

Pinder, G.F. (1973) A Galerkin-Finite Element Simulation of Groundwater Contamination on Long Island, New York. Water Resources Research, 9, 6: 1657-1669.

Schwartz, F.W. (1977) On Radioactive Waste Management: Model Analysis of a Proposed Site. Journal of Hydrology, 32; 257-277.

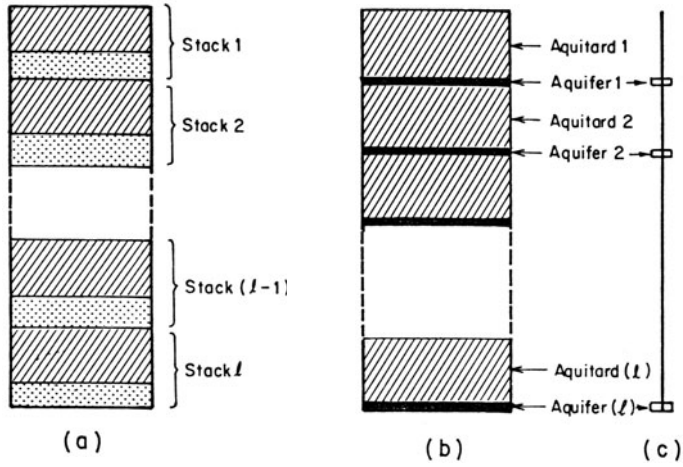


Figure 1 Multi-Layer System and its Model Representation. (a) Multi-Layer System; (b) Model Representation of Multi-Layer System; (c) Model Representation of Multi-Layer System at a General Point in the Area

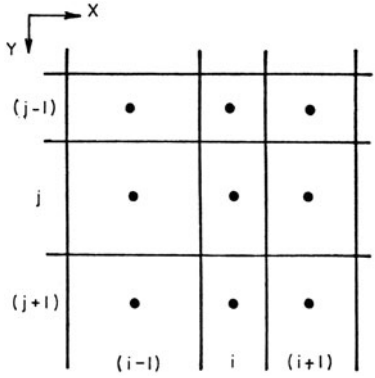


Figure 2a Non-Uniform Discretisation of Aquifers

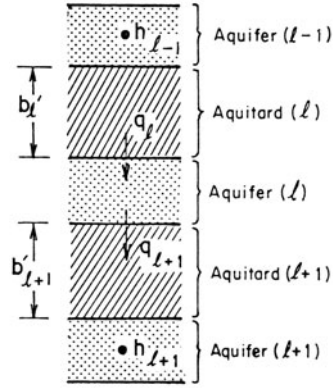


Figure 2b Estimation of Leakage Flux

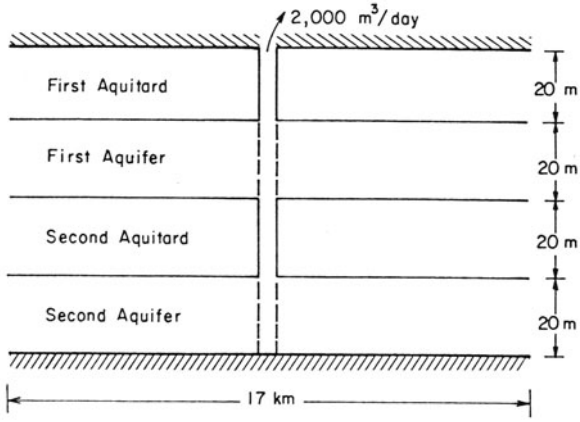


Figure 3 Aquifer System Used to Illustrate the Applicability of the Model

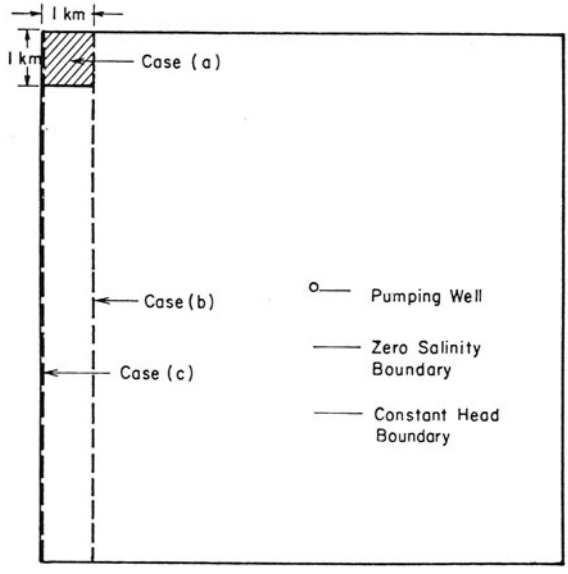


Figure 4 Plan View of the System with Source of Contamination. Case (a) - Connate Water Body in First Aquifer; Case (b) - River Acting as Line Source; Case (c) - Sea Acting as Line Source. Source Concentration = 10,000 $mg\ell^{-1}$.

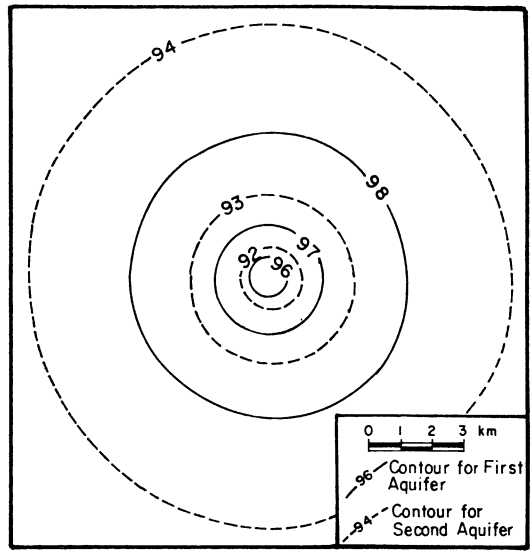


Figure 5 Piezometric Levels in Aquifers (in m. above reference plane)

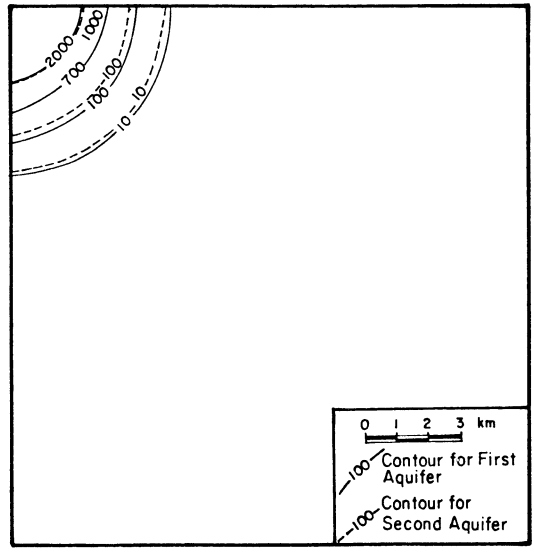


Figure 6 Chloride Distribution (in $mg\ell^{-1}$) in Aquifers after 20 Years of Simulation. Case (a)

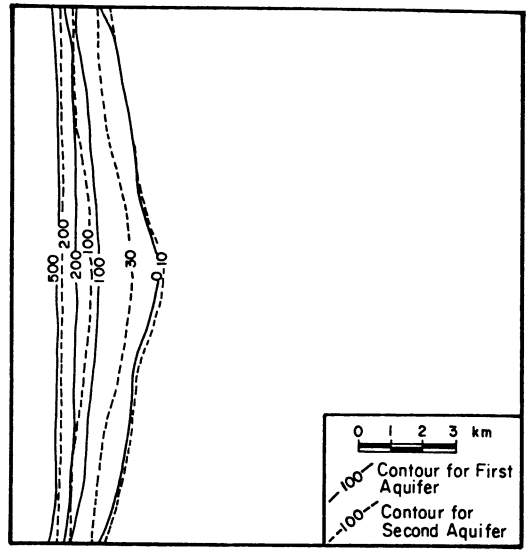


Figure 7 Chloride Distribution (in $mg\ell^{-1}$) in Aquifers after 20 Years of Simulation. Case (b)

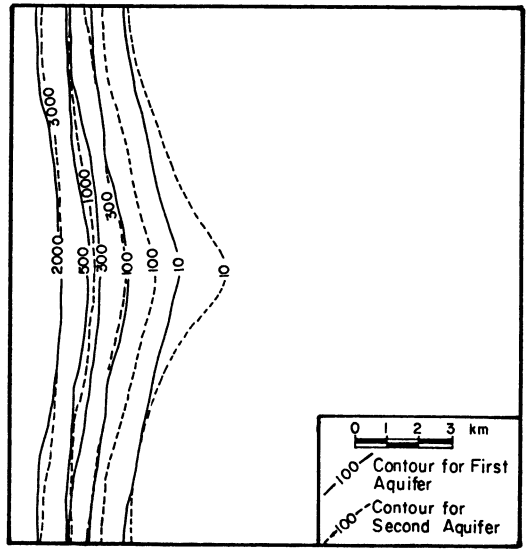


Figure 8 Chloride Distribution (in $mg\ell^{-1}$) in Aquifers after 20 Years of Simulation. Case (c)

MICROCOMPUTER APPLICATIONS TO DESIGN OF THE LOW HEAD PUMPED HYDRO SYSTEM AT RIVER SITE

Hitoshi Kinno

Emeritus Professor of The University of Tokushima (Japan)
1 The Grove, Golders Green, London N.W. 11, England

INTRODUCTION

Pumped hydro system has been used for long year period as energy storage system. However, it had gone toward using high head machine. On the contrary, low head one became to no use. It is caused to high head system has lower construction cost than low head one. But overall efficiency of conventional systems have never been improved more than 70 percent. This is due to that they have common two reservoir, namely upper reservoir and lower reservoir, where both static head of pump and turbine are same.

In order to improve overall efficiency, the author devised three reservoir system in where, for instance, pump head is half of turbine head. The idea was reported by Kinno (1979) and some applications to energy storage system at tidal power sites were reported by Kinno (1981) and (1982).

In this paper, the three reservoir system is applied for low head pumped hydro energy storage system at river site where overall efficiency is much larger than conventional one. Moreover, the system can be used for multipurpose as follows, a part or full river water stored to the highest reservoir using surplus cheap energy at nighttime can be used for sources of irrigation, drinking water, industry, etc, by only opening the gates mounted to both side dikes of the reservoir.

PRINCIPLE OF THE PUMPED HYDRO SYSTEM

Figure 1 shows a schematic diagram of the proposed pumped hydro system where (a) is top view and (b) is side view. When two dams are installed along a river where one is upstream dam and the other is downstream dam, three ponds appear naturally which consist of upstream pond with A Km^2 area, center pond called high pond with B Km^2 area and downstream pond.

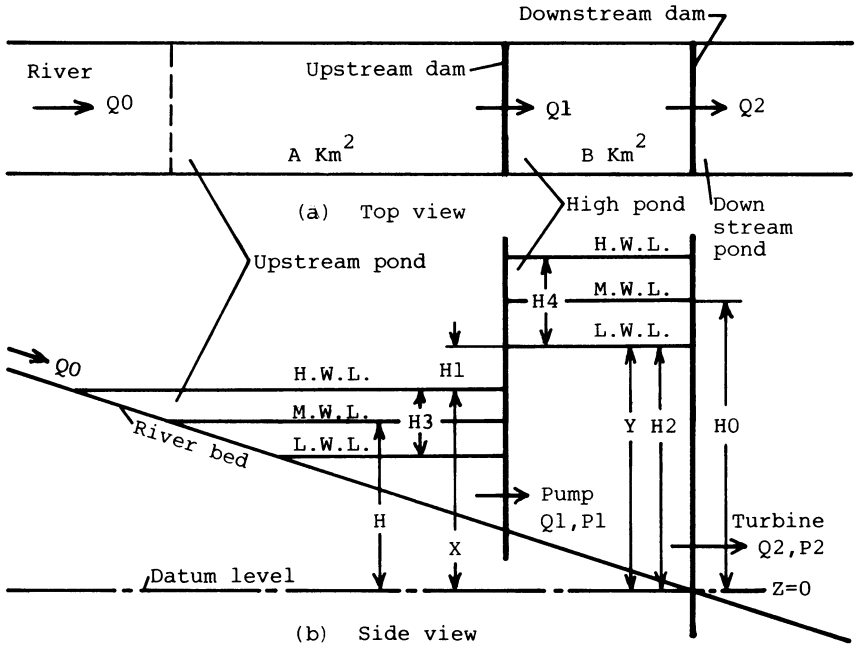


Figure 1 Schematic diagram of the proposed system

Let assume that the downstream pond's level can be controlled as to have uniform elevation, Z=0, which is used for the datum level of this system. Pumps are installed at bottom of the upstream dam and turbines are at bottom of the downstream dam. If inflow to the upstream pond is Q0 m^3/sec and pump is rest, water level of the upstream pond rises up to H.W.L..
At the nighttime, the pumps are driven by surplus energy come from grid, then high pond level rises up to H.W.L. while the water level of the upstream pond drops to L.W.L..
At the daytime, the turbine generate electric power to cover peak energy demand while water level of high pond drops to L.W.L. and water level of upstream pond rises up.

One of merits of this system is mean pump head,H0-H, is designed to about half of mean turbine head,H0; it makes extreme higher overall efficiency more than 130 %.

SYSTEM DESIGN

Basic system parameters

In order to design the system, basic system parameters have to be determined as follows:

Mean turbine head H0,m, is mean turbine head which has to be determined from the site situation in the first place.

For instance, if L.W.L. of high pond is designed as same level as ground level, the stored whole water in the high pond can be used for not only power generation but also as intake water to irrigation, drinking water, industry, etc.

Inflow to upstream pond Q0,m^3/sec, is inflow to upstream pond that is used to be taken to annual mean value.

Pump operation time T1,hr, is pump operation time. Pump start time is determined to 22.00 hrs in this paper, but it can be changed easily in program.

Turbine operation time T2,hr, is turbine operation time. Turbine start time is determined to 10.00 hrs in this paper, but it is changeable in program.

Reference head H,m, is mean water level of upstream pond that is taken to the reference head which has important role to the system efficiency. H is determined as function of H0 as H=K*H0, where K is one of basic system parameters.

Effective depth of upstream pond H3,m, is effective depth of upstream pond that is determined as function of H as H3=K1*H, where K1 is a basic system parameter.

Effective depth of high pond H4,m, is effective depth of high pond that is determined as H4=K1*H0.

Variables of the system
X is water level of upstream pond at transient. Y is water level of high pond at transient. Then pump head becomes as,

$$H1=Y-X \quad (1).$$

And turbine head is

$$H2=Y-Z \quad (2)$$

where Z=0.
F1 is mean pump efficiency shown by following empirical formula as,

$$F1=0.85-0.25*K1 \quad (3).$$

Samely F2 is mean turbine efficiency as,

$$F2=0.9-0.125*K1 \quad (4).$$

Area of upstream pond and downstream pond Area of upstream pond can be obtained from following equation as,

$$A=Q0*3600*(24-T1)/(H3*10\hat{}6) \quad (5)$$

where 10^6=10^6 in BASIC because H3 is determined by the inflow Q0 during (24-T1) hr.
Samely area of high pond can be introduced from following equation as,

$$B=Q0*3600*24/(H4*10^6) \quad (6)$$

because H4 is determined by one day's inflow which is pumped up to high pond.

Machine power Power of pump and turbine can be obtained from following equations as,

$$P1=\frac{H1*Q1*U}{101970*F1} \quad (7)$$

$$P2=\frac{H2*Q2*U*F2}{101970} \quad (8)$$

where P is power,MW_e, 1 MW_e=101970 Kgf-m/sec; H1 and H2 are head, m; Q1 and Q2 are discharge,m^3/sec; U is weight of water per unit volume,Kgf/m^3, which is 1000 for river water; F is machine efficiency and subscript 1 is pump and 2 is turbine, respectively.

Generally, low head pump and turbine used to adopt variable blade machine under constant speed. In these machine, constant power operation could be achieved under variable head. Then P1 and P2 have to be determined at M.W.L. of each pond and at each mean machine efficiency. Then P1 and P2 become as,

$$P1=(H0-H)*Q0*(24/T1)*1000/(101970*F1) \quad (9)$$

$$P2=H0*Q0*(24/T2)*1000*F2/101970 \quad (10).$$

For instance, if T1=T2=8 hr, mean discharge of pump and turbine become to three times of inflow.

Modelling of pump and turbine Mathematical models of pump and turbine can be obtained from ONE MW PLANE shown in Figure 2 which was reported by Kinno (1981). If maximum pump efficiency is 0.85 and P1=1MW, following equation can be obtained from equation (7) as,

$$H1*Q1=86.67 \quad (11)$$

which is a hyperbola. Pump model can be shown by the tangent to the hyperbola at H as,

$$Q1=86.67*(2*H-H1)*P1/(H*10^2) \quad (12).$$

Mathematical turbine model can be obtained from the ONE MW PLANE and equation (8) as,

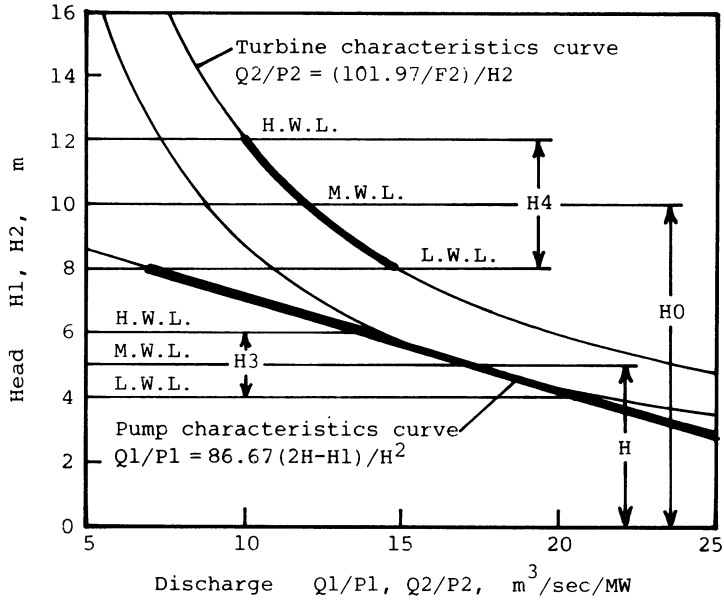


Figure 2 Mathematical models of pump and turbine

$$Q2=(101.97/F2)*P2/H2 \qquad (13).$$

Then the turbine model becomes a hyperbola itself at mean efficiency because H2*Q2=101.97/F2 can be obtained on the ONE MW PLANE.

Reservoir model Mathematical model of upstream pond reduces to obtaining water level change in a small time which is taken to 0.1 hr in this paper.
X1 and Y1 are water level change of upstream pond and high pond during 0.1 hr, respectively.
At pump operating phase,

$$X1=\frac{(Q0-Q1)*3600*0.1}{A*10\hat{}6}=0.00036*(Q0-Q1)/A \qquad (14)$$

$$Y1=0.00036*Q1/B \qquad (15).$$

At turbine operating phase,

$$X1=0.00036*Q0/A \qquad (16)$$

$$Y1=-0.00036*Q2/B \qquad (17).$$

At all machine stopping phase,

$$X1=0.00036*Q0/A \quad (18)$$

$$Y1=0 \quad (19)$$

Time limits of machine operation Pump starts at 22.00 hrs and stops when

$$Y>=H0+H4/2 \quad (20).$$

Turbine starts at 10.00 hrs and stops when

$$Y<H0-H4/2 \quad (21).$$

Thus H.W.L. and L.W.L. of the high pond are limited accurately, but no limit are given to upstream pond's level in program. T3 and T4 are actual operation time of pump and turbine under above limit of Y that can be computed from T3=T3+0.1 and T4= T4+0.1 at each step of I.

Pump input energy and turbine output energy E1 is pump input energy,MWH, in a day that can be obtained from following equation as,

$$E1=P1*T3 \quad (22).$$

Samely turbine output energy becomes as,

$$E2=P2*T4 \quad (23).$$

Overall efficiency F is overall efficiency which is

$$F=E2/E1 \quad (24).$$

Table 1 is a list of symbols appeared in this paper.

Table 1 Principal symbols

Symbol	Dimension	Description
A	Km^2	area of upstream pond
B	Km^2	area of high pond
C	hr	time convert variable
D	day	computing period control variable
D1	day	date
E1	MWH	pump input energy
E2	MWH	turbine output energy
F		overall efficiency
F1		mean pump efficiency
F2		mean turbine efficiency
G	hr	time
H	m	reference head
H0	m	rated turbine head (mean turbine head)
H1	m	pump head

H2	m	turbine head
H3	m	effective depth of upstream pond
H4	m	effective depth of high pond
I		loop control variable
J, J1		printout pitch control variables
K		reference head coefficient
K1		pond depth coefficient
N		loop control variable
P1	MW_e	pump input power
P2	MW_e	turbine output power
Q0	m^3/sec	inflow to upstream pond
Q1	m^3/sec	pump discharge
Q2	m^3/sec	turbine discharge
T	hr	time
T1	hr	specified pump operation time
T2	hr	specified turbine operation time
T3	hr	actual pump operation time
T4	hr	actual turbine operation time
U	Kgf/m^3	weight of water per unit volume
X, X2	m	water level of upstream pond above datum level
X1	m	small change of X in a small time, 0.1 hr
Y, Y2	m	water level of high pond above datum level
Y1	m	small change of Y in a small time, 0.1 hr
Z	m	datum level, Z=0

MICROCOMPUTER PROGRAM WRITTEN BY BASIC

Flow chart of the program

Figur 3 is the flow chart of microcomputer program. D and J are designate variables to computing period,day, and printout pitch, for instance, D=10 is "compute 10 days", J=1 is "printout pitch is J/10=0.1 hr". Usually, D<=10; J=1,5,10 and 240 are used. It is important that computing step is always 0.1 hr despite of any values of D and J.

Microcomputer program

Program 1 is the microcomputer program to obtain transients of D1, G, X, Y, H1, Q1, H2, Q2 and operating phase during D days; and each day's results of T3, T4, E1, E2 and F versus input data, H0, Q0, K, K1, T1 and T2 wher D1 is date and G is time, o'clock. Why continue ten days computation? It is due to following reason that the maximum and minimum limits of Y are designated but no limits of X, T3 and T4 are given except initial value of X. According to computing progress from D=0 to 10, T3=T1 and T4=T2 will be obtained, at which true initial value of X and the maximum and minimum water level of upstream pond can be obtained. Used microcomputer is TRS-80 III.

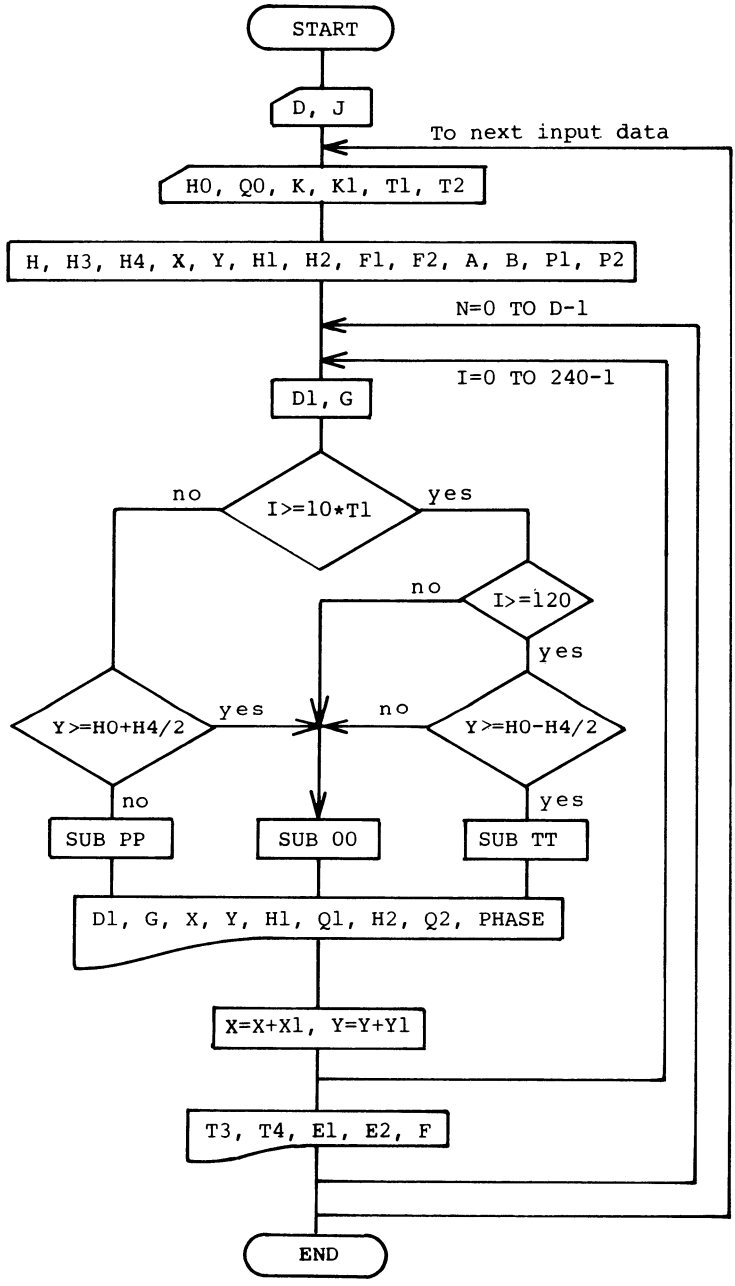


Figure 3 Flow chart of the computer program

Program 1

```
10 REM SOFT3 01-11-82 LOW HEAD PUMPED HYDRO SYSTEMS
20 DIM I(240)
30 READ D,J
40 READ H0,Q0,K,K1,T1,T2
50 IF H0=0 THEN 530
60 PRINT H0;Q0;K;K1;T1;T2
70 H=K*H0:H3=K1*H:H4=K1*H0:X=H+H3/2:X2=INT(X*100+0.5
)/100:Y=H0-H4/2:Y2=INT(Y*100+0.5)/100:H1=Y-X:Z=0:H2=
Y-Z
80 F1=0.85-0.25*K1:F2=0.9-0.125*K1
90 IF H1<=0 THEN 40
100 A=Q0*3600*(24-T1)/(H3*10[6):A=INT(A*100+0.5)/100
110 B=Q0*3600*24/(H4*10[6):B=INT(B*100+0.5)/100
120 P1=(H0-H)*Q0*(24/T1)*1000/(101970*F1):P1=INT(P1*
10+0.5)/10
130 P2=H0*Q0*(24/T2)*1000*F2/101970:P2=INT(P2*10+0.5
)/10
140 IF P1>=P2 THEN 40
150 LPRINT:LPRINT:LPRINT:LPRINT
160 LPRINT "-----------------------------------------
------------"
170 LPRINT "H0=";H0;TAB(9);"Q0=";Q0;TAB(18);"K=";K;T
AB(27);"K1=";K1;TAB(36);"T1=";T1;TAB(45);"T2=";T2
180 LPRINT "-----------------------------------------
------------"
190 LPRINT
200 LPRINT "COMPUTING  PERIOD,";D;"DAY";TAB(29);"PRI
NTOUT PITCH,";0.1*J;"HOUR"
210 LPRINT
220 LPRINT "H0=";H0;TAB(9);"A=";A;TAB(18);"H=";H;TAB
(27);"H0=";H0;TAB(36);"P1=";P1;TAB(47);"T1=";T1
230 LPRINT "Q0=";Q0;TAB(9);"B=";B;TAB(18);"H3=";H3;T
AB(27);"H4=";H4;TAB(36);"P2=";P2;TAB(47);"T2=";T2
240 LPRINT "-----------------------------------------
------------"
250 LPRINT
260 LPRINT "D1";TAB(4);"G";TAB(11);"X";TAB(17);"Y";T
AB(24);"H1";TAB(31);"Q1";TAB(38);"H2";TAB(44);"Q2";T
AB(50);"MC"
270 LPRINT
280 FOR N=0 TO D-1
290 IF N>=D THEN 40
300 T3=0:T4=0
310 FOR I=0 TO 239
320 T=N*24+I/10
330 D1=INT(T/24)+1:C=T+22:C1=INT(C/24)*24
340 G=INT((C-C1)*10+0.5)/10
350 IF I>=10*T1 THEN 380
360 IF Y>=H0+H4/2 GOSUB 670  :GOTO 430
```

```
370 GOSUB 550  :GOTO 430
380 IF I>=120 THEN 400
390 GOSUB 670  :GOTO 430
400 IF I>=240 THEN 440
410 IF Y>=H0-H4/2  GOSUB 760  :GOTO 430
420 GOSUB 670  :GOTO 430
430 X=X+X1:X2=INT(X*100+0.5)/100:Y=Y+Y1:Y2=INT(Y*100
+0.5)/100
440 NEXT I
450 T3=INT(T3*100+0.5)/100:T4=INT(T4*100+0.5)/100
460 E1=INT((P1*T3)*10+0.5)/10:E2=INT((P2*T4)*10+0.5)
/10
470 F=INT((E2/E1)*100+0.5)/100
480 LPRINT:LPRINT
490 LPRINT "T3=";T3;TAB(8);"HOUR";TAB(14);"E1=";E1;T
AB(24);"MWH"
500 LPRINT "T4=";T4;TAB(8);"HOUR";TAB(14);"E2=";E2;T
AB(24);"MWH";TAB(30);"OVERALL EFFI. F=";F
510 LPRINT:LPRINT
520 NEXT N
530 END
540 REM************************************************
550 REM SUBROUTINE PUMP
560 H1=INT((Y-X)*100+0.5)/100
570 Q1=INT((86.67*(2*H-H1)*P1/H[2)*10+0.5)/10
580 T3=T3+0.1
590 X1=0.00036*(Q0-Q1)/A
600 Y1=0.00036*Q1/B
610 J1=I-INT(I/J)*J
620 IF J1>0 THEN RETURN
630 LPRINT D1;TAB(2);G;TAB(9);X2;TAB(15);Y2;TAB(22);
H1;TAB(28);Q1;TAB(50);"PP"
640 RETURN
650 END
660 REM************************************************
670 REM SUBROUTINE 0000
680 X1=0.00036*Q0/A
690 Y1=0
700 J1=I-INT(I/J)*J
710 IF J1>0 THEN RETURN
720 LPRINT D1;TAB(2);G;TAB(9);X2;TAB(15);Y2;TAB(50);
"00"
730 RETURN
740 END
750 REM************************************************
760 REM SUBROUTINE TURBINE
770 H2=INT(Y*100+0.5)/100:Q2=INT(((101.97/F2)*P2/H2)
*10+0.5)/10
780 T4=T4+0.1
790 X1=0.00036*Q0/A:Y1=-0.00036*Q2/B
800 J1=I-INT(I/J)*J
810 IF J1>0 THEN RETURN
```

```
820 LPRINT D1;TAB(2);G;TAB(9);X2;TAB(15);Y2;TAB(35);
H2;TAB(42);Q2;TAB(50);"TT"
830 RETURN
840 END
850 REM***********************************************
860 REM DATA
870 REM DATA D,J
880 DATA 5,10
890 REM DATA H0,Q0,K,K1,T1,T2
900 DATA 10,300,0.5,0.1,8,8
910 DATA 10,300,0.5,0.2,8,8
920 DATA 10,300,0.5,0.3,8,8
930 DATA 10,300,0.5,0.4,8,8
940 DATA 10,300,0.5,0.5,8,8
950 DATA 10,300,0.5,0.6,8,8
2000 DATA 0,0,0,0,0,0
```

NUMERICAL EXAMPLES

Input data

As show in the last part of program 1, input data have to be given as D=5, J=10; and H0=10m, Q0=300 m^3/sec, K=0.5, K1=0.4, T1=8 and T2=8. We can change any above input data, however in program 1, K1 is only changed as K1=0.1 to 0.6 for another fixed input data in order to inspect the effect of K1.

Output data

Table 2 is an output data printed on a line printer in where a part of D1=1 (the first day) and full part of D1=3 (the third day) are shown, this is steady state condition, because T3=T1=8 and T4=T2=8 are obtained, and at same time, E1=470.4 MWH, E2=600 MWH and F=1.28 are obtained.

Effects of K, K1, T1 and T2

Figure 4 to Figure 7 are computed results when K, K1, T1 and T2 are changed one each respectively in their practical range, but another input data are all fixed. Among these four figures, effect of K is largest for overall efficiency and the second is K1 but T1 and T2 give no effect.

However, the larger value of K makes the higher level of upstream pond and the smaller pump power; the former may makes flood caused by high backwater to upstream basin, and the latter makes little absorption of surplus energy as energy storage system . As a common data, K=0.5, K1=0.4, T1=8 and T2=8 are used in figures 4 to 7, they are useful data at starting point of the systems design.

Machine number and unit power

It is recommended that pump and turbine have same number of unit, because both operating units number can be decreased proportionally to inflow reduction than Q0 so as to keep systems similarity. For example, if total power of pump and turbine were

Table 2 An example of output data

H0= 10 Q0= 300 K= .5 K1= .4 T1= 8 T2= 8

COMPUTING PERIOD, 3 DAY PRINTOUT PITCH, 1 HOUR

H0= 10 A= 8.64 H= 5 H0= 10 P1= 58.8 T1= 8

Q0= 300 B= 6.48 H3= 2 H4= 4 P2= 75 T2= 8

D1	G	X	Y	H1	Q1	H2	Q2	MC
1	22	6	8	2	1630.8			PP
1	23	5.5	8.84	3.34	1357.6			PP
1	0	5.1	9.53	4.43	1135.4			PP
1	1	4.79	10.12	5.33	952			PP
1	2	4.55	10.61	6.06	803.2			PP
			middle		cut			
3	22	6.07	7.99	1.92	1647.1			PP
3	23	5.57	8.84	3.27	1371.9			PP
3	0	5.16	9.54	4.38	1145.6			PP
3	1	4.85	10.13	5.28	962.2			PP
3	2	4.6	10.62	6.02	811.3			PP
3	3	4.41	11.04	6.63	687			PP
3	4	4.27	11.4	7.13	585			PP
3	5	4.17	11.7	7.53	503.5			PP
3	6	4.1	11.96					00
3	7	4.22	11.96					00
3	8	4.35	11.96					00
3	9	4.47	11.96					00
3	10	4.6	11.96			11.96	752.3	TT
3	11	4.72	11.54			11.54	779.7	TT
3	12	4.85	11.1			11.1	810.6	TT
3	13	4.97	10.64			10.64	845.6	TT
3	14	5.1	10.16			10.16	885.6	TT
3	15	5.22	9.66			9.66	931.4	TT
3	16	5.35	9.13			9.13	985.5	TT
3	17	5.47	8.56			8.56	1051.1	TT
3	18	5.6	7.96					00
3	19	5.72	7.96					00
3	20	5.85	7.96					00
3	21	5.97	7.96					00

T3= 8 HOUR E1= 470.4 MWH

T4= 8 HOUR E2= 600 MWH OVERALL EFFI. F= 1.28

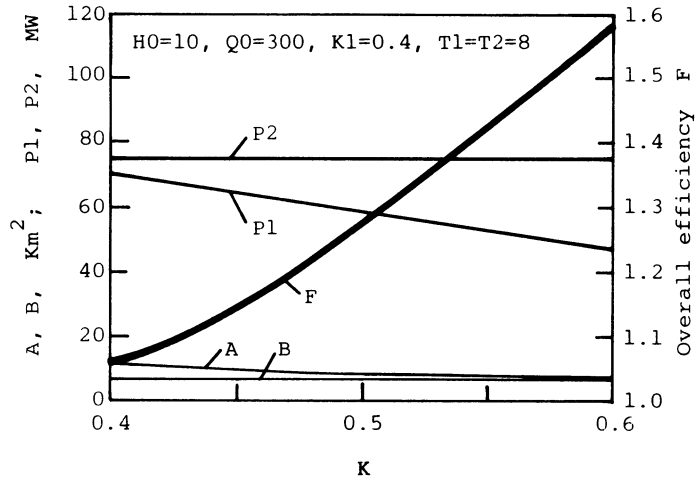


Figure 4 Effect of K to the system when another input data are fixed

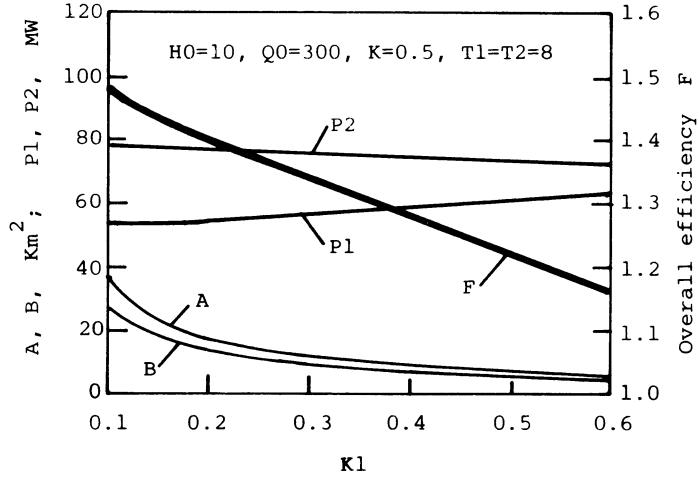


Figure 5 Effect of K1 to the system when another input data are fixed

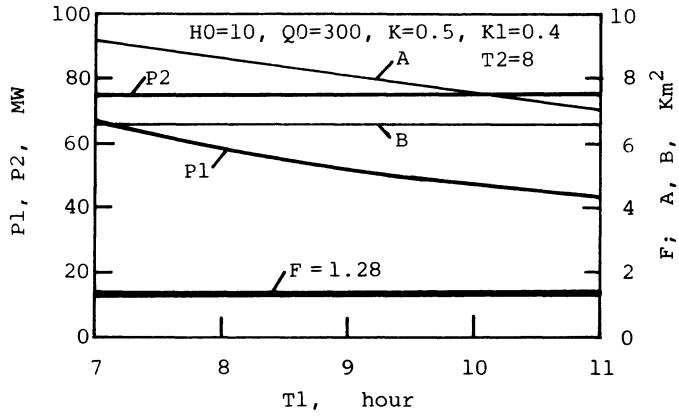


Figure 6 Effect of T1 to the system when another input data are fixed

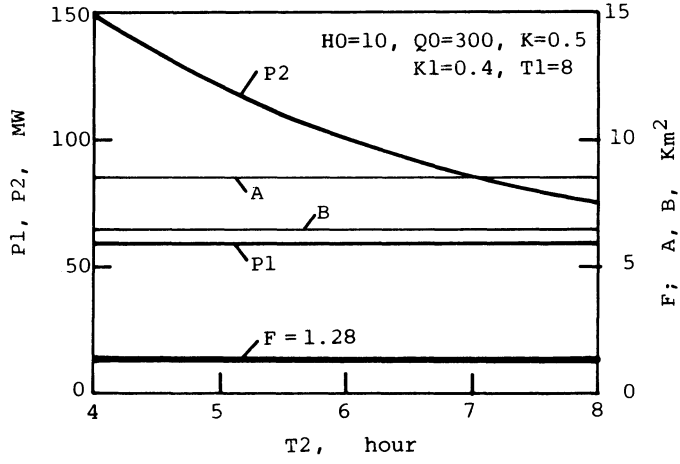


Figure 7 Effect of T2 to the system wnen another input data are fixed

obtained from table 2 as P1 = 58.8 MW ≒ 60 MW and P2 = 75 MW, machine number × unit power are determined as, pump: 5×12 MW, turbine: 5×15 MW or pump: 15×4 MW, turbine: 15×5 MW.

FLOOD PROTECTION

When inflow increases more than annual mean value Q0, it is important that water level of upstream pond does not rises up a

limited head, for instance, H.W.L. in figure 1. Pump discharge is designed as to have 3*Q0 at rated head, 5.4*Q0 at 2 m head from table 2 and 6*Q0 at zero meter head from mathematical pump model, these pump capacity could be kept European rivers in safty. Remember that high pond level can be easily droped down by opening the gates mounted downstream dam whether or not turbine operation.

It is interest that this systems can be changed to the continuous operating system which was reported by Kinno at IWRA (1982), whenever it is needed, in where turbines could drive the pump without any outside energy, because output power of the turbine always larger than the pump input power even at flood condition. Then the excess water flowed into the upstream pond caused larger inflow than Q0 can be excluded to down stream side through pump, turbine and gate using the self operating ability at out of scheduled time of energy storage, if the flood is not so large.

CONCLUSIONS

Low head pumped hydro energy storage systems are presented in where usually negligible low head energy at lower river is boosted up by pump using surplus energy to a effective head for turbine. In the system, pump head is taken to about half of turbine head, it makes extra large systems efficiency more than 130 percent.

Moreover, a part or whole water pumped up using surplus energy to high pond which is higher than ground level can be used for sources of irrigation, drinking water, industry, etc, by only opening the gate mounted on the dikes of high pond instead of power generation. Thus power generation and intake water are convertible from each zero to maximum.

REFERENCES

Kinno, H. (1979) Pumped-storage Battery with 100 Percent Efficiency. Proc. of the 14th IECEC, Boston, Vol. 1, 414-418.

Kinno, H. (1981) CAD of the Self-retiming Mechanism at Large Tidal Power Sites. Proc. of the Second International Conf. on Engineering Software, London, 496-511.

Kinno, H. (1982) A microcomputer Program for Design of the Self Retiming Mechanics of Large Tidal Power Sites. Proc. of the 4th International Conf. on Finite Elements in Water Resourses, Hannover, 8:3-12.

Kinno, H. (1982) Design of the Convex Reservoir on a River. Selected Papers of the IVth World Congress of IWRA on Water for Human Consumption, Buenos Aires, Vol. 2, D94-63.

COMPUTER PROGRAMS FOR RISK OPTIMIZATION IN WATER RESOURCES

Slobodan P. Simonović

Institute for the Development of Water Resources
Jaroslav Černi, Belgrade, Yugoslavia

INTRODUCTION

This paper presents reliability programming approach in water resources systems management. Reliability programming is one stochastic technique which directly incorporates risk into the optimization. The comparison of deterministic problem, stochastic problem with risk defined a priori, and stochastic problem with risk determined explicitly is presented in the paper. The reliability programming model is shown to be nonlinear and could be split into two models: search model and linear programming model, which could be solved serially as part of an overall algorithm. By using a reliability programming approach, the concept of reliability/risk is included directly in the optimization. Paper presents detailed explanation of general solution algorithm and five computer programs developed for the purpose of water resources system management (single multipurpose reservoir; multiple-multipurpose reservoir system; etc.).

RELIABILITY PROGRAM FORMULATION

Reliability programming is a subcategory of stochastic programming that generally deals with the theory and methods of incorporating stochastic variations into a mathematical programming problem. Consider the linear programming model (LP) in matrix notation:

$$\text{maximize} \quad z = c^t x \tag{1}$$

subject to

$$Ax \leq b \tag{2}$$

$$x \geq 0 \tag{3}$$

in which: c = vector of the objective function coefficients; b = resource vector; x = vector of decision variables; A = matrix of coefficients; and t = superscript to denote the transpose of vector c. In this model, the parameters in set (A,b,c) are given and known numbers, and it is required to determine an optimal decision vector x subject to the specified constraints (deterministic problem). If the elements in set (A,b,c) are stochastic, then stochastic variations are introduced into the programming problem through variations in Ω, in which Ω denotes the vector with elements (A,b,c).

An important case of stochastic programming arises when we assume chance-constraints, i.e., constraints that are not expected to be always satisfied, but only with given probabilities. A simple LP model is said to be chance-constrained programming (CCP) if its linear constraints (2) are associated with a set of probability measures (called reliabilities or

risks) indicating the extent of the violation of the constraints. Assuming that only the elements b_i of the resource vector b are random and mutually independent, consider a simple version of CCP. In this chance-constrained LP, a tolerance level in terms of probability measures, one for each probabilistic constraint, is preassigned by the decision maker. Consider the LP system (1)-(3) and let a_i denote the ith column of matrix A; x the column vector with n elements; and b_i the element of resource vector b ($i = 1,2,\ldots,m$). It is assumed that each b_i is random and has a nonnegative domain and cumulative distribution function:

$$F(a_i^t x) = P\{b_i \leq a_i^t x\} = 1 - \alpha_i \tag{4}$$

$$0 \leq \alpha_i \leq 1 \tag{5}$$

Then a simple chance-constrained LP model can be written as:

$$\text{maximize} \quad z = c^t x, \tag{6}$$

subject to

$$P\{b_i \leq a_i^t x\} \geq \alpha_i, \qquad i = 1,2,\ldots,m \tag{7}$$

$$x \geq 0 \tag{8}$$

$$0 \leq \alpha_i \leq 1, \qquad i = 1,2,\ldots,m \tag{9}$$

in which α_i is a tolerance level vector and m is the number of constraints. CCP problem formulation requires preassigning the values of tolerance level vector. Instead of discussing the question of a satisfactory choice of α_i, a new viewpoint of relia-

bility programming (RP) can be considered.

RP formulation is an alternative way of incorporating the tolerance level vector into the optimization. This can be done by introducing the utility function

$$U = U(\alpha_i, z) \tag{10}$$

of the decision maker with reliabilities (α_i) and profits ($z = c^t x$) as two arguments satisfying the conditions

$$\frac{\partial U}{\partial z} > 0, \quad \frac{\partial U}{\partial \alpha_i} > 0, \quad \frac{d\alpha_i}{dz} < 0 \tag{11}$$

which guarantee the attainment of the U function's maximum value in cases where the U function is concave in its arguments. One possible choice of the utility function satisfying condition (11) is:

$$U = w_1 \sum_{j=1}^{n} c_j x_j + w_2 \sum_{i=1}^{m} \log \alpha_i \tag{12}$$

in which w_1 and w_2 are scalar nonnegative constant weights. The transformed programming problem, now called RP problem, may be presented as:

$$\max U = w_1 c^t x + w_2 \sum_{i=1}^{m} \log \alpha_i \tag{13}$$

subject to

$$P\{b_i \geq a_i^t x\} \geq \alpha_i, \qquad i = 1,2,\ldots m \tag{14}$$

$$x \geq 0 \tag{15}$$

$$0 \leq \alpha_i \leq 1, \qquad i = 1,2,\ldots m \tag{16}$$

A very important difference between CCP and RP formulation can be noted. A set of optimal α_i's is here solved along with the optimal value of decision variables $x_j (j = 1,2,\ldots,n)$. This is different from the CCP approach, in which the set of tolerance measures α_i's is preassigned by the decision maker.

Two basic questions in formulating reservoir management as a reliability programming problem are to be considered. First is the question of estimation and numerical computation of the distribution function of the random variable (in case of reservoir management, random inflow). The second question is how to estimate and compute the utility function (12), especially its second part.

GENERAL SOLUTION ALGORITHM

A two-level algorithm was developed to solve the reliability program, (13) through (16). The first level, a search algorithm, is used for the optimization of reliability values. The second level, an optimization scheme, is used for the optimization of water resources system performances. A figurative scheme of the two-level solution algorithm is presented in Fig. 1. An adequate multidimensional search technique is chosen for the first level and one of the mathematical programming techniques is chosen for the second level. To overcome the complexity of reliability program formulation, (13) through (16), the concavity theorem of the objective function is introduced. Let $\bar{\alpha}_i$ be such that $F_t(\alpha_i)$ is concave for $\bar{\alpha}_i \leq \alpha_i < 1$. Let $U^o(\alpha_i)$ be the optimum of the objective function (13) in the program (13) through

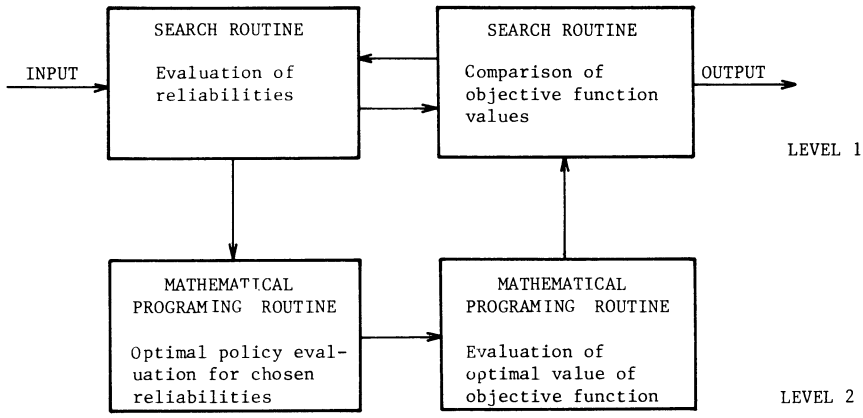


Figure 1. Two-level solution algorithm

(16) for a fixed values of α_i. For each set of α_i, $u^o(\alpha_i)$ has the following property.

Theorem 1: The function $u^o(\alpha_i)$ is concave in the domain $D(\alpha_i)$, where $\bar{\alpha}_i \leq \alpha_i < 1$. The proof of Theorem 1 is a simple expansion of the proof given by Simonović (1981). The reliability program (13) through (16) can be now rewritten as:

$$\underset{\{\alpha_i\}}{\text{maximize}} \quad U^o(\alpha_i) \tag{17}$$

$$\alpha_i \in D \qquad i=1,2,\ldots,m \tag{18}$$

In view of Theorem 1, program (17) and (18) is concave.

The reliability programming approach is developed by Simonović (1981) and applied to the single multipurpose reservoir planning, Simonović and Marino (1980), and multiple - multipurpose reservoir planning, Simonović and Marino (1982). Also, considerable research was done in determining conditioned distribution functions, Simonović and Marino (1981).

COMPUTER PROGRAMS

For the purpose of reliability program application in reservoir operation five computer programs were developed:

CDFC -- computer program for fitting the monthly reservoir inflows with conditioned gamma distribution. The program is written in FORTRAN and contains 408 instructions. It is very efficient, requiring 0.12 CPU minutes on Burroughs 6700.

CONVO-- computer program for convoluting the distribution functions of random reservoir inflows. The program is written in FORTRAN and contains 262 instructions. One run for convoluting the CDF's for twelve months requires 0.03 CPU minutes on Burroughs 6700.

RPORC-- computer program for single-multipurpose--reservoir management via reliability programming. This computer program was designed with two-dimensional Fibonaccian search. The program is written in FORTRAN and contains 860 instructions. For accuracy of solution

equal to 0.05, this program requires 10.47 CPU minutes on Burroughs 6700.

RPORCS-- computer program for single-multipurpose--reservoir management via reliability programming. This computer program was designed with Box search algorithm. The program is written in FORTRAN and contains 980 instructions. For accuracy of solution equal to 0.05, this program requires 1.38 CPU minutes on Burroughs 6700.

SYSRE-- computer program for multiple-multipurpose--reservoir-system management via reliability programming. This computer program is written in FORTRAN and contains 1740 instructions. It requires a computer with large-core memory. For a system of three reservoirs and an accuracy of 0.05, this program requires 6.52 CPU seconds on CDC 7600.

This section contains a detailed analysis of the structure of the developed computer programs.

CDFC-- Computer program for fitting the monthly reservoir inflows with conditioned gamma distribution. The program contains a main program and five subroutines, connected as shown in Fig. 2, for the purpose of fitting the historical data with conditioned gamma distribution. The program was designed to handle historical-data series of monthly flows. The computational procedure is described in Simonović and Marino (1981).

The main program reads the monthly flow data and then

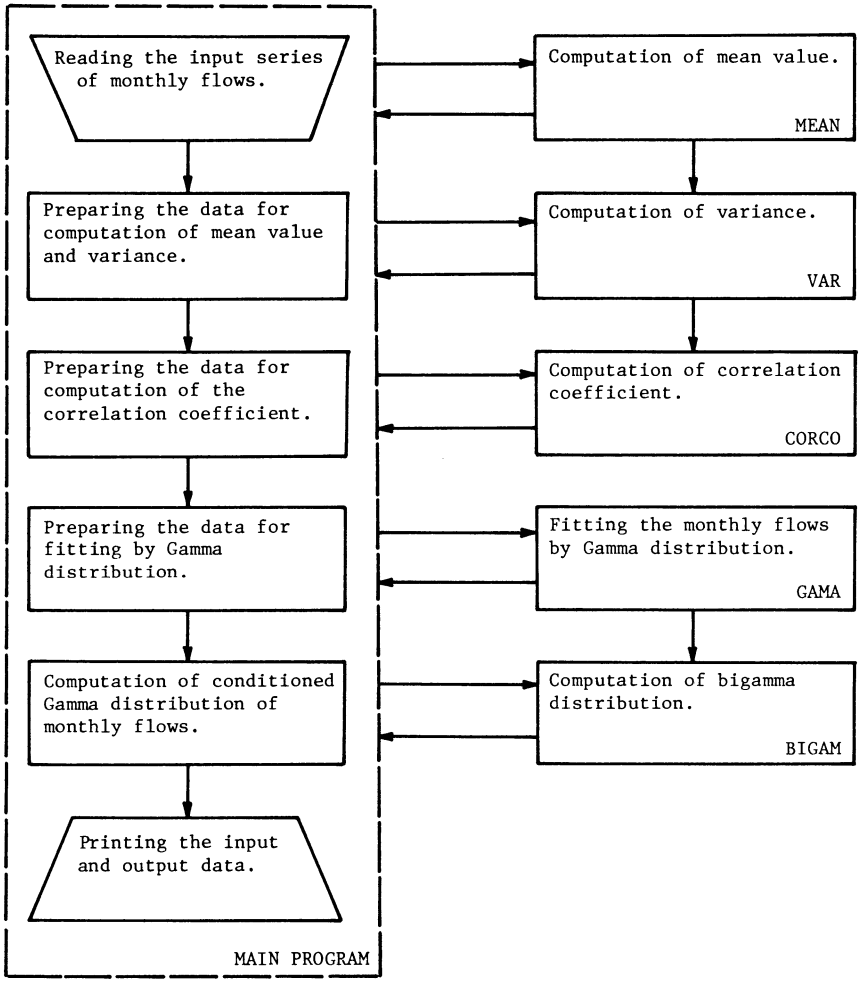


Figure 2. Structure of the computer program for determination of the conditioned CDF's of monthly flows.

calls subroutines MEAN and VAR, which compute, respectively, mean value and variance for each month. For any month, the correlation coefficient is computed between that month and the month before. For this purpose, the main program calls subroutine CORCO. Now, by calling subroutine GAMA, the data for each month are fitted with gamma distributions. Finally, by subroutine GAMA, the bivariate gamma function is computed. In the main program, conditioned CDF's are computed and input and output data are printed.

The input data contain the historical series of monthly flows and number of years of observations. The output block contains the conditioned distribution functions of monthly flows.

CONVO--Computer program for convoluting the distribution functions of random reservoir inflows. This program reads the distribution functions of inflows, and then, using the iterative convolution algorithm Simonović (1979) adds them cumulatively. To illustrate how the program works, consider the hydrological year with known distribution functions of monthly flows. Program CONVO computes, first, the sum of distributions of October and November flows, then adds to this sum December flows, then to this sum January flows, and so on.

The program contains a main program and two subroutines. The main program reads the input CDF's of monthly flows. Then using subroutine PREV, the CDF's are transformed into probability density functions (PDF's). PDF's are then used in subroutine CONVZ to find the convoluted sum of two PDF's. Finally, in the main program, convoluted PDF's are transformed into

CDF's and printed out.

The input data contain the CDF's of monthly flows. The output set contains convoluted CDF's of monthly flows. The CONVO program structure is shown in Fig.3.

RPORC--Computer program for single-multipurpose--reservoir management via reliability programming. This program uses an algorithm combined with two--dimensional Fibonacci search and linear programming. The program structure is based on a procedure described in Simonović and Marino (1980). The program contains a main program and five subroutines, as shown in Fig. 4.

The main program after reading the input data, performs a search in α directions and subroutine FIB performs a search in β directions. For every value of α and β, subroutine CCP finds the value of the objective function. To obtain the optimal objective function, subroutine CCP uses subroutine LPS, which is the simple linear program, and subroutines PREV and CONVZ, explained in the description of the CONVO computer program. When the search accuracy criterion is satisfied, the whole procedure is concluded and the main program is used to print the output data.

RPORCS--Computer program for single-multipurpose--reservoir management by reliability programming. This program is designed with Box complex search algorithm, which appears to be much more efficient than Fibonacci search algorithm (especially in cases of more than two dimensions).

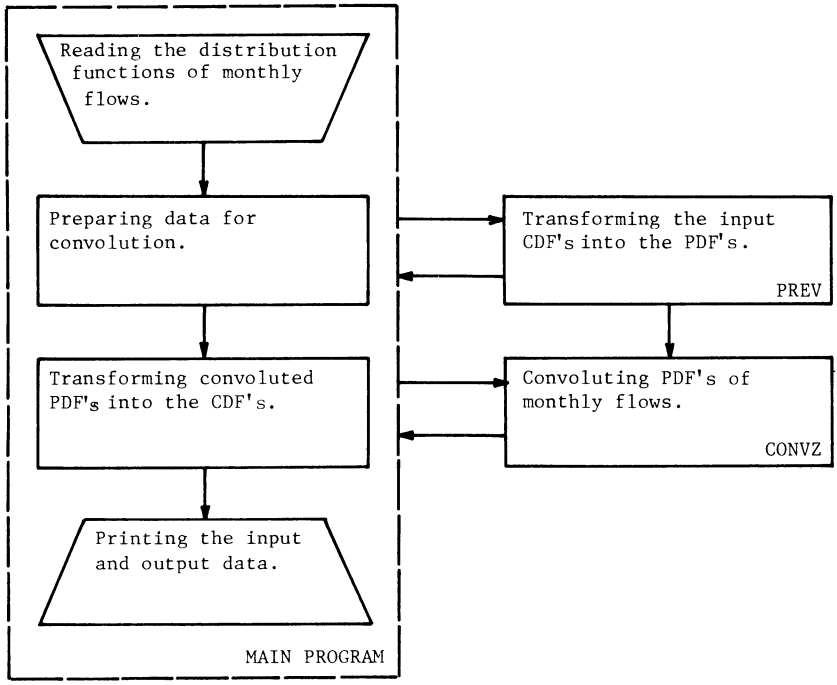


Figure 3. Structure of the computer program for convolution of monthly inflow CDF's.

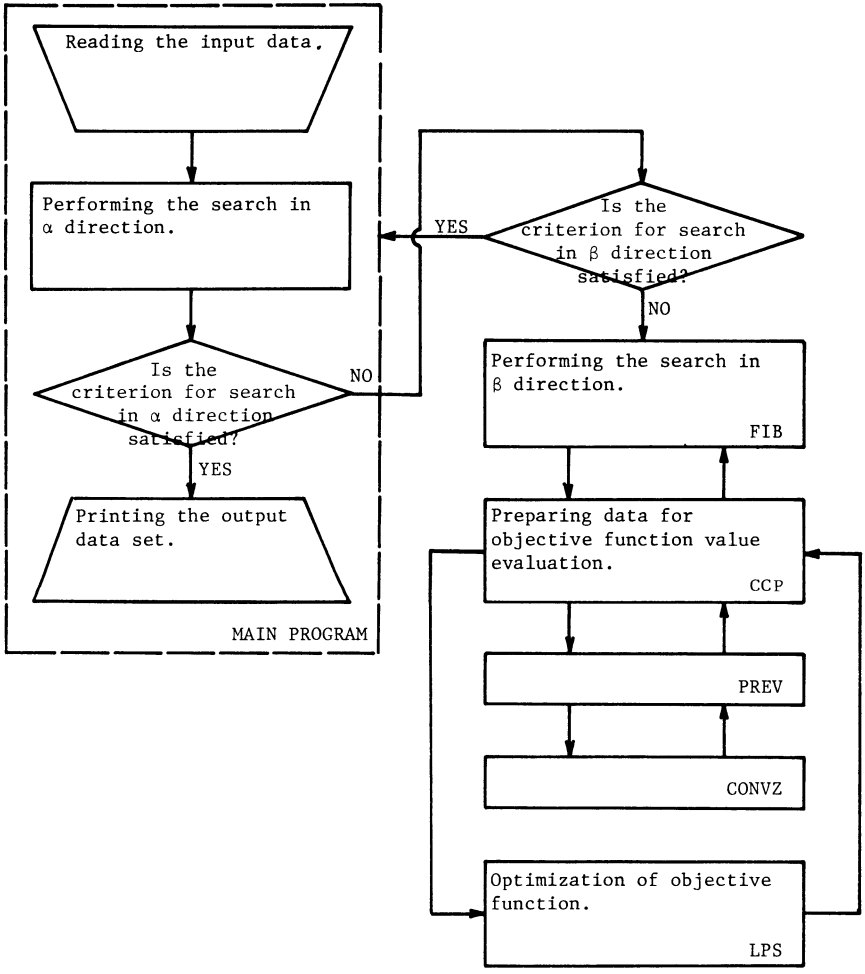


Figure 4. Structure of the computer program for single multipurpose reservoir management by reliability programming.

The RPORCS program contains a main program and eight subroutines. The main program reads the input data and prints the output data on the end. The computation is performed by the subroutines. Four subroutines are the parts of the search block: CONSC--subroutine for coordination of search routines; CHECK--subroutine for checking all points against search constraints; CENTR--subroutine for calculating the centroid of points; and CONST--subroutine that specifies explicit and implicit search constraint limits. Other subroutines are the same as in program RPORC: CCP, PREV, CONVZ, and LPS.

The four subroutines in search block search for optimal reliability values. They use subroutines, CCP, PREV, CONVZ, and LPS for objective function evaluation and computation of optimal values of decision variables. When the search criterion is satisfied, the procedure concludes and prints the output data. The complex structure of RPORCS program is shown in Fig. 5.

SYSRE-Computer program for multiple-purpose-reservoir system management via reliability programming. This program is constructed of a main program and a set of eight subroutines. The program design is based on the theory presented in Simonović and Marino (1982). The main algorithm has two levels: search and mathematical programming. The search level, based on Box algorithm, contains four subroutines: CONSC, CHECK, CENTR, and CONST. The other part contains subroutines CCP--subroutine that prepares LP tableau and coordinates the other mathematical programming routines; LPSIM--subroutine that solves the LP with bounded variables; OF--subroutine for computing the objective

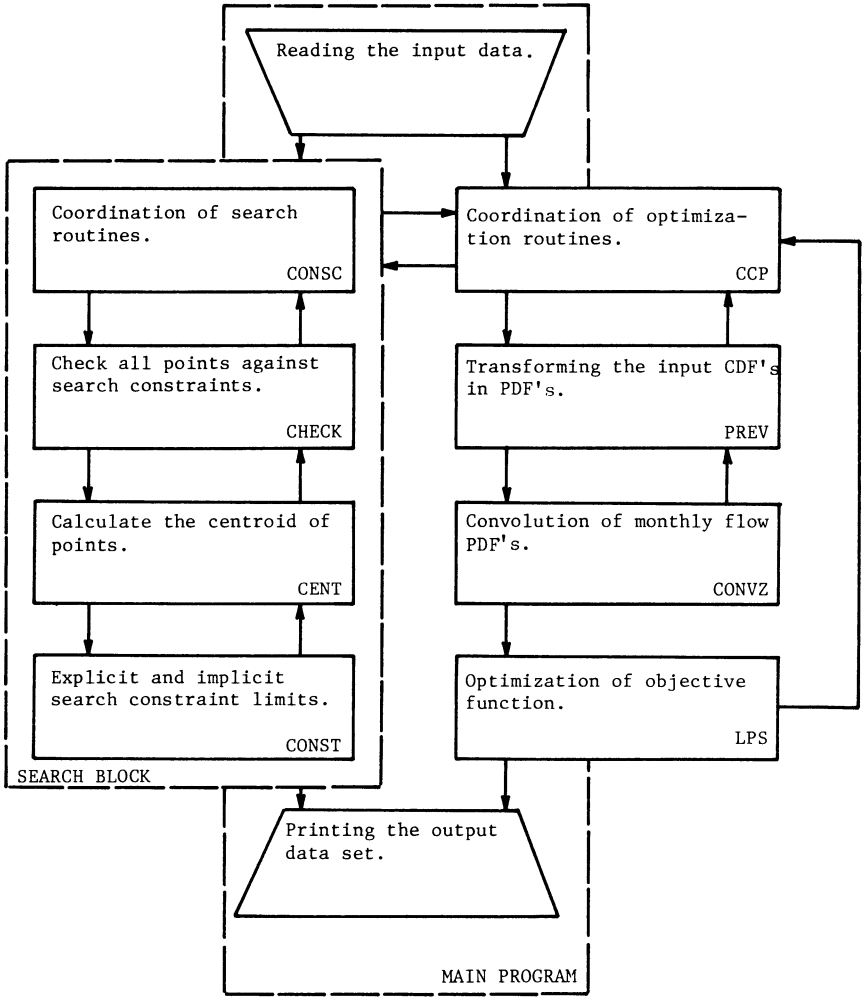


Figure 5. Structure of the computer program for single multipurpose reservoir management by reliability programming.

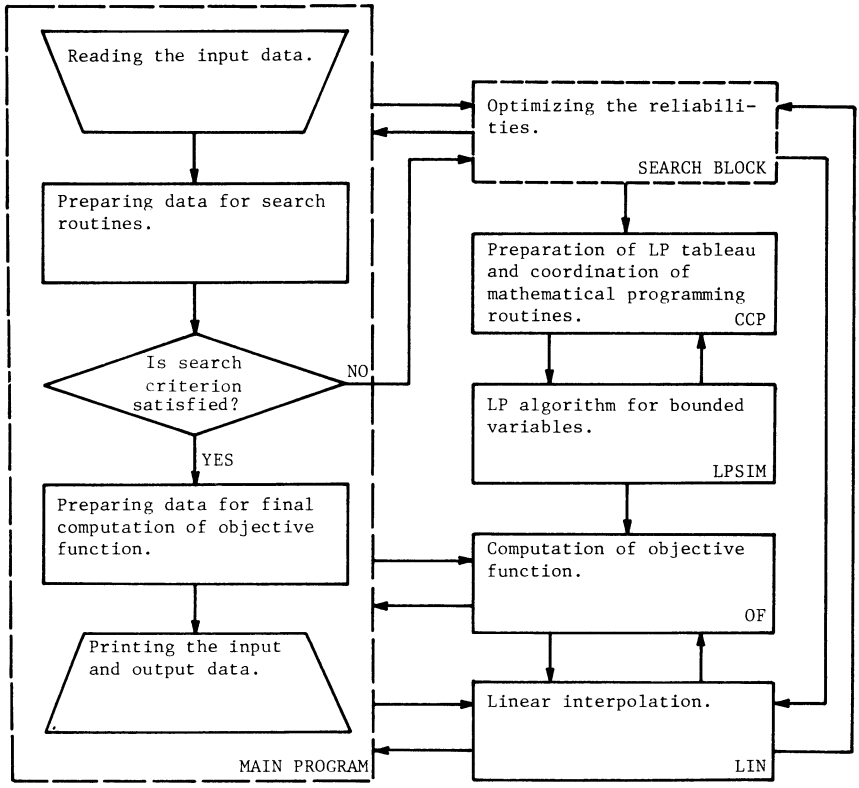


Figure 6. Structure of the computer program for multiple-multipurpose reservoir system management via reliability programming.

function; and LIN--subroutine for linear interpolation (helping subroutine used in few places in the program). The structure of SYSRE is shown in Fig. 6.

The SYSRE program requires as input the output results of CDFC and CONVO programs. This is why all three programs must be considered a package for the multiple-multipurpose-reservoir-system management.

CONCLUSIONS

This paper presents an introduction of the reliability programming approach to water resources systems management. The approach, in allowing reliabilities to be considered as decision variables, explicitly considers the trade-off between benefits and risk. To emphasize the major advantages of the reliability program comparisons between deterministic linear program, chance-constrained program and reliability program have been presented. Finaly, five computer programs developed for the purpose of water resources system management are presented in the paper.

REFERENCES

Simonović, S.P. (1979) Two-step algorithm for design--stage long-term control of a multipurpose reservoir. Advances in Water Resources 2:47-49. March.

Simonović, S.P. and M.A. Marino (1980) Reliability programming in reservoir management, 1. Single multipurpose reservoir. Water Resources Research 16(5):844-848.

Simonović, S.P. and M.A. Marino (1981). The role of unconditioned and conditioned cumulative distribution functions in reservoir reliability programming. Journal of Hydrology 52(3/4):219-228.

Simonović, S.P. (1981) Reliability programming in reservoir management. Ph.D. dissertation, University of California, Davis.

Simonović, S.P. and M.A. Marino (1982) Reliability programming in reservoir management, 3. System of multipurpose reservoirs. Water Resources Research (in print).

A REAL TIME RIVER FLOW FORECASTING SYSTEM

P J Wolfenden Systems Designer, Data Processing Department
J R Douglas Area Hydrologist, Severn Area Resources Unit
Severn Trent Water Authority
Birmingham

ABSTRACT

This paper describes the design and operation of a real time river flow forecasting system developed by Severn-Trent Water Authority. Although the overall system is outlined, emphasis is placed on the software developed and implemented on an IBM Series/1 minicomputer.

Objectives realised in the design of the software were twofold. Firstly, the controlling software offers facilities such as the real time acquisition of hydrometric data and co-ordination of all system action and interaction between users internal to and remote from flow forecasting centres. Secondly, application based software has been established, the main emphasis of which has been to apply conceptual and mathematical models in order that timing and severity of flood conditions may be predicted. Forecasts of future river conditions are produced routinely for a large number of sites throughout the basins of the Severn and Trent.

INTRODUCTION

During periods of fair weather, it can be very pleasant to live on a river bank. During dry weather it can be highly advantageous to farm land adjacent to a river. But following heavy rainfall and rapid snowmelt, the river's benefits can, quite literally, be swamped by the cost and inconvenience of flooding. The Severn-Trent Water Authority has a policy of making all reasonable efforts to mitigate the effects of flooding.

Wherever it has been calculated to be both cost effective and practicable, flood mitigation is achieved by engineering works. In many cases, however, flood protection would not be economically viable, and in an attempt to keep flood damage

and losses to a minimum, a flood warning service is provided to many communities. The average annual saving to the community has been estimated by Chatterton et.al.(1979) to be in excess of £0.75 million. To maximise the benefits of such a service, flood warnings have to be provided to those at risk sufficiently early to allow belongings to be moved upstairs or livestock to be evacuated. On the other hand false alarms can quickly reduce the credibility of the warning service. Great reliance is placed, therefore, on the availability of accurate forecasts of future river levels at all points where flooding is likely to occur and to cause damage. A flood forecasting system has been evolving over the past 15 years: the present state of this system is the subject of this paper.

Providing the data

Widespread flooding in the 1960's and in particular in July 1968 triggered considerable investment in field instrumentation. Before then, information on river levels could only be obtained by people visiting gauges and reporting back by telephone. The new instruments can be interrogated remotely over telephone lines enabling data to be collected without visiting the site. The network of gauges shown in Figure 1 has been established over the intervening 14 years. From this network the state of all the significant rivers in the Severn and Trent catchments can now be monitored. A network of 76 raingauges over the 20 000 km^2 enables a reasonably accurate assessment of rainfall amounts over any part of the area to be made. A certain amount of climatological information is also available by telemetry from a limited network of automatic weather stations.

The network of telemetry gauges is connected through the telephone system, and all gauges can be interrogated at any time from any standard telephone. Until recently all data were collected manually. Dialling a network of up to 100 gauges, listening to and logging the transmitted data was becoming an increasingly onerous task as the network expanded. Collecting data through the night was particularly unpopular.

In April 1982 a telemetry 'scanner' was commissioned. This device, based around a Plessey 'Miproc' mini computer and DTS Telegen 4000 front end processor allows data to be collected from the field by machine (Figure 2). The amount of data collected can be suited to climatological and catchment conditions using a flexible system of polling option control programs.

Producing forecasts of river level

Collecting the data is but the start of the forecaster's task. In the simplest river systems, it is possible to relate

downstream flooding to trigger levels at an upstream site. Warning systems based on this principle have been in existence on the upper reaches of the rivers Severn and Vyrnwy in Powys for more than a decade. Provided that the upstream site is well chosen, and that the correct threshold levels are identified, these systems work well in their limited areas.

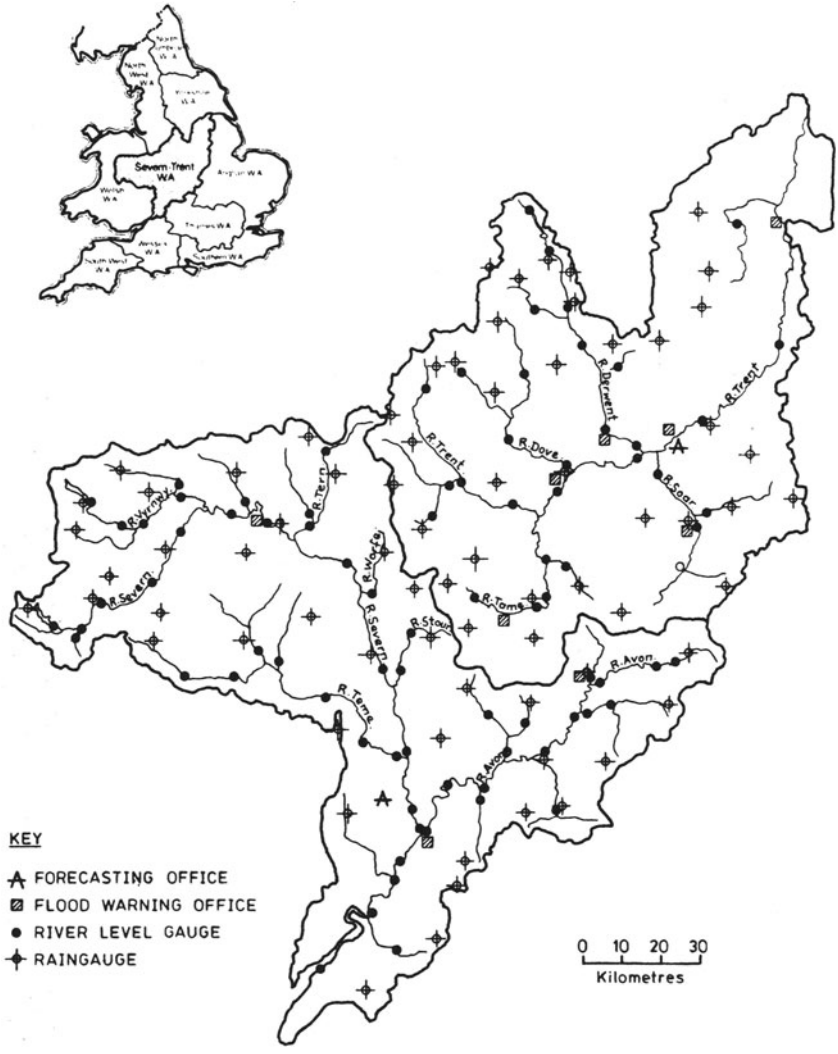


Figure 1 Location map showing flood warning scheme telemetry network

The problems begin when the river system upstream of the settlement at risk cannot be described as simple: a typical example is the town of Shrewsbury. Built on a meander of the Severn, Shrewsbury is just 25 km downstream of the confluence of the Severn and its major tributary, the Vyrnwy (Figure 1). A flood in Shrewsbury depends on the magnitude of the flood in both tributaries; on the coincidence of the flood peaks in each river, and on the amount of water held in storage on the extensive flood plain upstream of the town. Shrewsbury is typical: our forefathers built their settlements in flood plains, at river confluences, on tidal estuaries - in almost all the situations that make flood forecasting difficult.

The advent of the computer has facilitated the development of mathematical models capable of describing the complex response of a river and its catchment to a rainfall event. Use of suitable models enables the flood forecaster to make full use not only of the data available to him in real time during a flood event, but also the accumulation of historic data which may be used in model calibration. A wide range of models is available. These fall into two broad categories, firstly models which convert rainfall into runoff, and secondly models which translate a runoff hydrograph down the river system. Although these types of models have existed for several years, they are not widely used for flood forecasting. The main reason for this is not a shortcoming in the models, but rather the difficulty of using them in the forecasting situation, where time is always short and manpower hard pressed. To facilitate their use in real time by forecasters rather than computer personnel, the interactive computer system which is the subject of this paper has been established. This enables the forecaster to produce model-based forecasts simply and quickly with a minimum of training in the use of the computer. The system runs on an IBM Series/1 minicomputer.

Operating the Service

Because floods are relatively rare events, it is not worthwhile paying hydrologists to man an office 24 hours a day. However, since almost all major floods occur overnight, especially during bank holiday periods (or so it seems!), the forecasting service is often operated from a duty officer's home. One of the requirements of the computer system was that it should be just as accessible from a remote portable terminal as in the office. In addition to the forecaster requiring access to the system from a remote terminal, flood warning staff in the Authority's eight Divisions need to have access to the forecast results. They are also able to communicate via portable terminals though their use of the system is restricted to its display facilities.

COMPUTER SYSTEM DESIGN

General concepts and objectives

As outlined above, and more fully in previous publications (Manley et. al. (1980), and Jones, (1980)), the flow forecasting system operates around two minicomputers (see Figure 2). The first, a Plessey-Miproc (hereafter referred to as the 'scanner') is used to interrogate rainfall and river level stations. Transference of data from remote field stations is achieved via standard telephone lines, at intervals of time ranging from 20 minutes to 24 hours. All data are stored initially on a 'prime' file which contains time, date and reading for the last 24 scans for up to 200 stations. Data older than this are archived on a fixed exchangeable disc. The scanner software provides a system in its own right, by not only collecting the required data, but also providing staff with formatted reports of river level, flow and rainfall values. This facility is available to users both internal to and remote from the forecasting office via portable terminals.

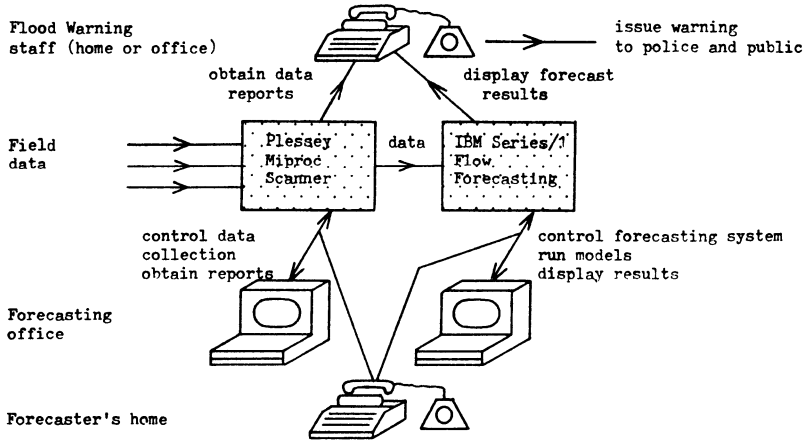


Figure 2 Diagrammatic representation of the forecasting hardware

Although an integral part of the overall flow forecasting system, emphasis is placed in this publication on the real time software developed on the second minicomputer, an IBM Series/1. As far as this part of the system is concerned, the scanner merely acts as a collector and provider of raw data. The objectives of the software to be designed and implemented on the Series/1 were:-

1. to provide the real time control of the system 24 hours per day

2. to allow communication between the scanner and Series/1 to facilitate transfer of gauge data

3. to allow communication between users at and remote from the forecasting office, such that they may access the system using portable terminals

4. to allow existing forecasting software to be incorporated into the real time system, such that changes to programs were kept to a minimum

5. to provide a log of all normal and error activity.

Above all, the main aim was to keep the software as simple as possible for the end user to understand and operate.

Major design considerations

The design of the flow forecasting system reflected the constraints and limitations imposed by the minicomputer. The Series/1 used was a 4955 Model E 16 bit processor architecture with a total core size of 256 K bytes, operating under Realtime Programming System (RPS) Version 4.0. Memory is effectively divided into two types of partition: a split instruction - data system partition (occupying approximately 142 K bytes) and two user partitions (occupying a total of 114 K bytes). Any one partition cannot exceed 64 K bytes of directly addressable primary storage.

A major concern throughout the design phase was that of deciding on a suitable technique to overcome the problem of fitting a large number of programs (of the order of 250 K bytes) into the user partitions such that 64 K bytes was not exceeded. Of the options tested and available, a decision was taken to divide a single taskset (i.e. a collection of programs and data executing in one partition) into multiple segments, and to retain only certain of these in primary storage all the time. As the most significant factor was a concern for processor storage, disk overlays, rather than storage overlays or user transients, were believed to be the best solution.

Two important factors emerge from the single taskset and disk overlay design. Firstly, the self containment of all programs and data operating in one partition allows other systems to run concomitantly in the second user partition. Such systems may be either completely separate from or directly associated

with the flow forecasting system (for example, a tidally influenced reaches model which is queued from the main taskset to execute in the second partition). Secondly, because disk overlay programs take the longest amount of time to execute, and speed is an important factor in real time design, access times to disk were minimised by coding these programs in 15 K byte overlays which are called into physical storage for relatively long time steps. In other words disk access was kept to a minimum.

Programs were coded in either Fortran IV or assembler language. The latter was used exclusively for terminal manager, line and timer tasks, whilst all other controlling and application software was written in Fortran (Figure 3).

All communication internal to the system was accomplished through the use of data queues, parameter lists, WAIT/POST macros and global areas. The synchronisation of task execution was achieved by defining events relevant to those tasks. Use of this method allowed flexibility and the running of several tasks at the same time. This maximises processor usage and is known as multitasking.

Data queues were used as a means of communication between tasks operating asynchronously. Storage queues, which were small in size (between 5 and 10 elements) were used because prompt responses were necessary and because storage was allocated dynamically from the taskset's partition. All queues were managed by prefixing them with three control words. These comprised a terminal identifier (to route messages to the appropriate channel), a 'control' character (used to identify initial breakin, to control the format for paged output and for certain error detections) and an identifier indicating which application program was executing.

SYSTEM OUTLINE

The system outline is shown diagrammatically in Figure 3. All three line tasks and terminal manager task monitor messages routed between a dedicated terminal (a Newbury VDU) and two portable terminals (Texas Silent 700s) and the main control task (FFCTRL). This flexibility in accessing the system allows the hydrologist, flood duty officer or Divisional flood warning staff remote from the forecasting office to obtain information on current and future river conditions provided by the application programs (FF01 to FF09) via standard telephone lines.

The main controlling software for the entire system is provided by the program, FFCTRL. This routine:

- controls the main queues on which data are routed between end user and application programs

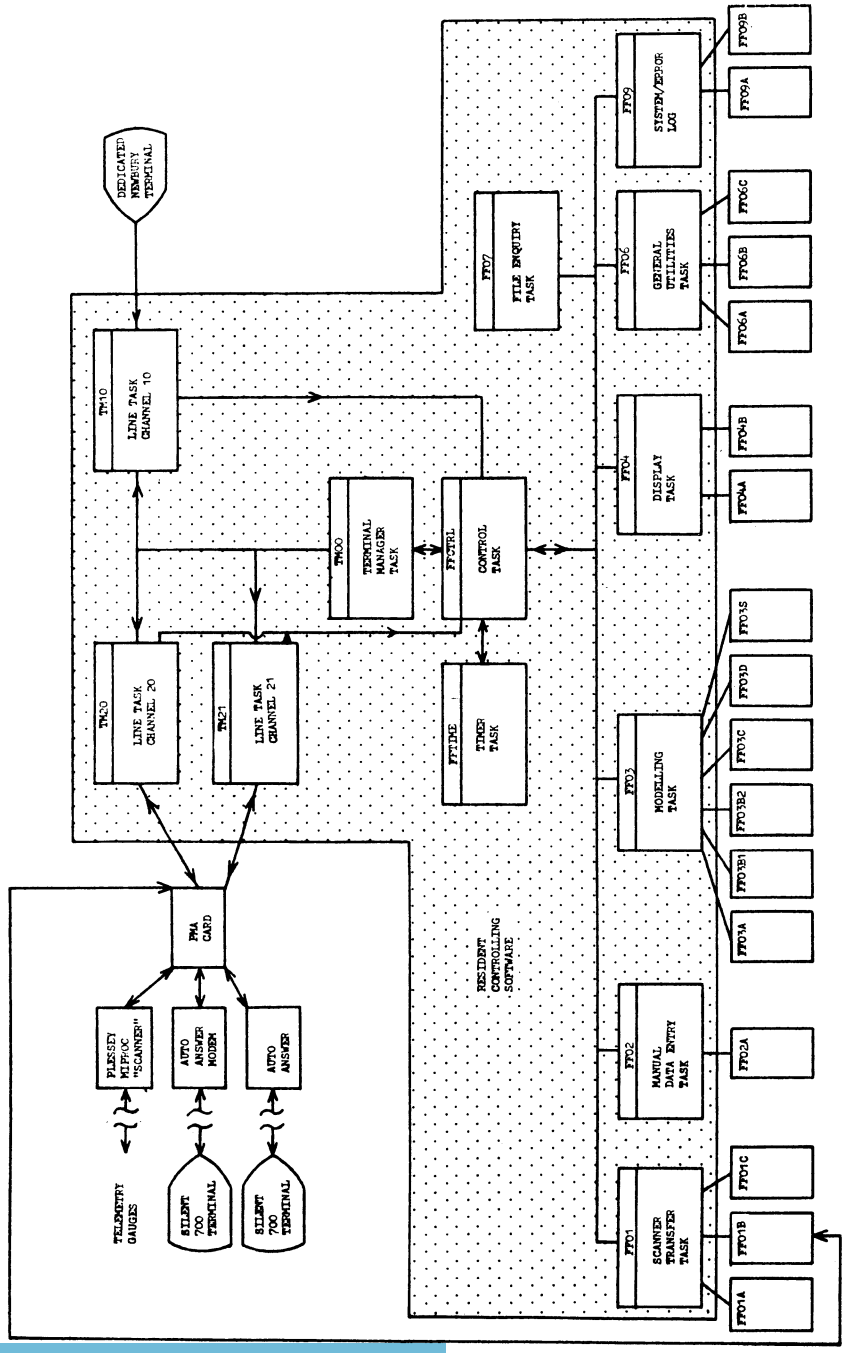


Figure 3 Overall design of forecasting system software

- provides a menu service and status report to the user indicating which programs are currently running

- provides error, validation and warning messages

- handles all internal communications associated with timer services, activity and error logs

- controls the setting and unsetting of major global flags

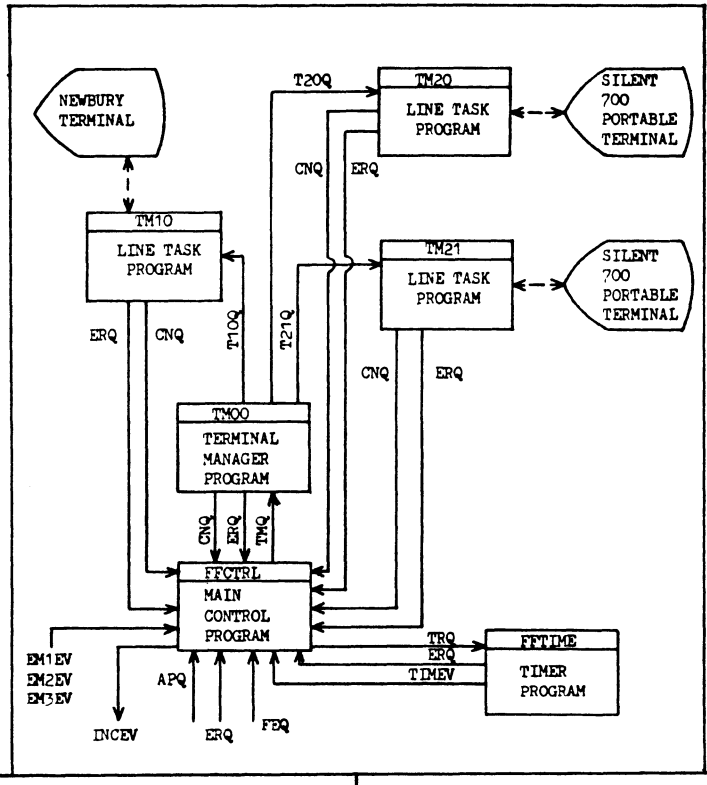
Timer services are controlled via the assembler routine, FFTIME (Figure 3). At pre-selected intervals of time an event is posted to the main control task which in turn attaches the scanner transfer task, FF01. This initiates the transfer of gauge data from scanner to Series/1. An interactive routine is additionally built in to FFTIME to allow the user not only to set up time related variables, but also to determine the extent and range of data to be transferred from the scanner.

Severe restrictions in memory size necessitate that only one application program may be called in to resident storage at any one time. This is achieved via the root segment of each program, which is permanently resident and which calls in other overlays as required. The one exception to this is the file enquiry program, FF07, which remains resident all the time, providing users with a 'bulletin board' of river flow information.

An integral part of the system is the controlling of all detectable errors. All non zero return codes or errors detected when reading or writing to files are flagged and sent via the error queue to be written as unformatted records on an activity/error log file. Information held on this file includes the terminal identifier, the program, time of error, internal code, return code or record number, hexadecimal completion codes and status bytes relating to external communications failures as well as a description of the internal code. Additionally, a message is issued from the control task to the user informing him that an error has occurred and that the offending program has been stopped. Certain types of error are also flagged via the second control word on the queue, CNQ, linking FFCTRL to TM00 (Figure 4). These may occur when starting the line tasks from TM00 or when starting the line tasks from TM00 or when specific line errors are detected. Depending on the seriousness of error, the application program running will be automatically stopped and an attempt made to reset and restart a line.

The system uses eleven main files, the data/information on them being held in either binary or character format. All

files are accessed directly using the record number as the key. Security copies of the main data files are presently taken manually on a twice weekly basis, although it is proposed that this function should be performed automatically as part of the overall system requirements.



QUEUE OR EVENT	DESCRIPTION (Used to ... ...
CNQ	Pass incoming messages to FFCTRL.
TMQ	Pass outgoing messages to TM00.
T10Q, T20Q and T21Q	Pass outgoing messages to TM10, TM20 and TM 21.
APQ	Handle outgoing messages from application programs to FFCTRL.
FEQ	Handle outgoing messages from the file enquiry program to FFCTRL.

TRQ	Handle question mode messages from FFCTRL to FFTIME.
ERQ	Handle all internal error messages to FFCTRL.
TIMEV	Posted to FFCTRL to initiate scanner transfer program.
INCEV	Posted to awaiting application program.
EM1EV EM2EV EM3EV	Posted to FFCTRL on detection of errors.

Figure 4 Controlling software highlighting the major internal communication events and queues

The controlling software

This section outlines in more detail the structure and operation of the controlling software. All programs execute in a similar manner. They are designed around primary and secondary waitlist statements which contain a series of event parameters. Whenever one of these events is satisfied, a single thread of execution (or task) takes place.

On initial start of the taskset, the control program (FFCTRL) attaches the timer, terminal manager (which in turn starts the line task programs) and system/error log program FF09 (see Figure 3). The latter produces a hardcopy listing of all activity and errors from the previous run and re-initialises the error log file. All controlling software is now set in a wait status.

All incoming messages to the system (whether initial break-in or user dial up, or replies to interactive program questions) are routed via the line task programs on the queue CNQ to FFCTRL (see Figure 4). In a similar manner all outgoing messages are funnelled via FFCTRL on TMQ to the terminal manager task, which in turn acts as a router to the appropriate line task programs (via T10Q, T20Q and T21Q).

Communication between the main control task and application programs is synchronised via the event, INCEV. This is posted to the relevant program awaiting this event, indicating that an incoming message can be processed. The queues APQ and FEQ (see Figure 4) control outgoing messages from application programs and the file enquiry program respectively. The timer program FFTIME, uses a logical asynchronous timer, which permits a task to be executed with the time operation running concurrently. At selected intervals of time determined by an interactive routine with the user (via the queue TRQ), the event TIMEV is posted to FFCTRL. Although presently this only initiates the running of the scanner transfer program, future enhancements to the system will enable other programs to be run automatically in real time mode. Finally, all detectable errors from any program are sent to the queue ERQ which the main control task handles. At the same time the events EM1EV, EM2EV and EM3EV are posted which cause suitable error/warning messages to be routed from FFCTRL to the appropriate channel.

The application programs

The controlling software allows the running of application programs (shown as FF01 to FF09 in Figure 3) which manipulate the data to produce and display flood forecasts. These programs can be broadly divided into four groups;

1. data input programs,

2. data processing and modelling programs,

3. display and output routines,

4. utility programs

which are described in the following sections.

Data Input programs Two programs are available to enter data into the system. Most data are collected automatically by the scanner and transferred to the Flow Forecasting System by the scanner data transfer program, FF01. This is run automatically by the system at predetermined intervals, and collects all fresh data from the scanner. FF01 can be run on demand. Data can also be collected manually as described earlier, and any such figures may be entered into the forecasting system using an interactive manual data entry program, FF02. Gauge data are held on a direct access file: each gauge has a rolling storage holding the most recent 31 readings from the gauge.

Data processing and modelling programs A suite of programs has been assembled which performs all of the analysis of the data, converting raw raingauge and river level readings into meaningful forecasts. Before performing any modelling calculations, data are subjected to a series of preliminary operations:

1. readings from each gauge are sorted into chronological order.

2. quality control routines check for continuity between successive readings: doubtful continuity is queried during the interactive quality control session.

3. regular hourly data are produced from the spot gauge readings by linear interpolation.

4. catchment average data are produced for climatic variables such as rainfall, temperature and wind speed, from the available gauge data. Data from as many as five raingauges may be used to make the best possible estimate of catchment rainfall although very few of the catchment areas contain more than one raingauge.

5. catchment and river reach outflow data are computed from river level gauge readings where these are available. However, gauges only exist at the outfalls of approximately 30% of the catchments and 60% of the river reaches.

6. forecast rainfall data are interactively entered.

Once the preliminaries have been completed, the forecasting models can be run. Two models are used in the forecasting system; a 'rainfall-runoff' model and a flow routing model. Both of these models operate on a continuous basis, so that the initial values of all state variables (such as soil moisture deficit, and groundwater storage in the rainfall-runoff models and channel and flood plain storages in the flow routing model) are those saved at the termination of the previous model run.

The rainfall-runoff model used is a conceptual model describing in mathematical terms all of the significant features of the hydrological cycle. It is a development of the model described by Douglas (1974), and contains the same runoff generating mechanism as that used in the Isolated Events Model described in the Flood Studies Report (NERC 1975). This is a non-linear relationship between the runoff percentage and the soil moisture deficit. The model also considers interception, percolation to groundwater, baseflow generation and the evapotranspiration processes. When appropriate a snowpack is modelled and snowmelt predicted as a function of climatological variables as described by Jackson (1978).

The flow routing model used is of the Muskingum type, where outflow from a river reach is determined from reach inflow and from the storage in the reach. The model has sufficient complexity to describe the behaviour of static storage areas which fill up once flood defences are overtopped, as well as describing the flow along the flood plain of a river in flood. The speed of propagation of the flood wave down a reach is variable and depends on the magnitude of the inflow to the reach. The model is similar in many ways to that described by Price (1977).

All the data used in the model computations, as well as gauge details, and parameter values for the models are stored in a suite of direct access disk files. To reduce program execution times, the bulk data are held in binary format, whilst parametric information is held in the less efficient character form, for convenience of file use.

The suite of processing and modelling programs are all part of the application task FF03, which contains six disk overlays. These overlays are called into core sequentially, thus minimising disk access time.

Display and Output Routines Two programs have been written to allow the forecaster access to the forecasts. The first, FF04, is a general purpose display system, which produces hydrograph plots for any of up to 100 sites, or multi-site

plots for any of 30 predefined groups of four sites, within the catchment. The display system will also display the raw gauge data, echoing a facility available on the scanner, and allow the user to post messages on the 'bulletin board'. Plotted output is produced using lineprinter symbols, so that it is immaterial whether the forecaster is working in the office or from a remote terminal. Summary output is available from a second program, FF07, the file enquiry task, which unlike the other application programs, is a resident part of the system. Whatever other program is running, it is always possible to run this program to obtain general catchment information and to read messages from the 'bulletin board'.

Utility programs A program, FF06, is available to enable interactive editing of parameter and gauge details files. This utility also allows the contents of the binary data files to be printed.

System activity and error occurrences are logged by the system as described earlier. A utility program, FF09, can be run on demand to produce a listing of activity and error information.

SUMMARY

The system described above has been developed over the past three years. Parts of the system have been 'live' since the autumn of 1981, although the calibrated models have only been used as a forecasting tool since the beginning of 1983. The performance both of the system and of the models is being monitored - early results are encouraging.

ACKNOWLEDGEMENTS

The authors would like to thank the Director of Operations of the Severn-Trent Water Authority for giving permission to publish this paper. They are also indebted to R G Sharp, T R Wood and A B Haynes for their support and encouragement throughout the project and to their colleagues, particularly Mrs L M Hill, for their contributions to the system development.

The views expressed in this paper are those of the authors, and do not necessarily represent those of the Authority.

BIBLIOGRAPHY

Chatterton J B, Pirt J and Wood T R (1979) The benefits of flood forecasting. J.Instn.Wat.Engrs.Scient. 33 no. 3.

Douglas J R (1974) Conceptual modelling in hydrology. Inst.Hydrol.,Wallingford. Rep.24.

Jackson M C (1978) The influence of snowmelt on flood flows in rivers. J.Instn.Wat.Engrs.Scient. 32 no.6.

Jones H H (1980) An overview of hydrological forecasting in a multi-functional Water Authority. Proc.Oxford Symp. IASH Publ.no.129.

Manley R E, Douglas J R and Pirt J (1980) Conceptual models in a flow forecasting system. Proc.Oxford Symp. IAHS Publ.no.129.

Natural Environment Research Council (1975) Flood Studies Report. NERC, London.

Price R K (1977) FLOUT - A river catchment flood method. Hydraulics Research Station, Wallingford. IT68.

SOFTWARE DESIGN FOR A VERSATILE FLOOD FREQUENCY ANALYSIS

Krishan P. Singh

Illinois State Water Survey, Champaign, Illinois, U.S.A.

INTRODUCTION

A new flood frequency methodology has been developed and computerized. It detects objectively the outliers and inliers at various significance levels and modifies them if needed. Six significance levels, 1 through 6 (corresponding to outlier-inlier probability pairs 0.01, 0.99; 0.05, 0.95; 0.1, 0.9; 0.2, 0.8; 0.3, 0.7; and 0.4, 0.6), are defined in addition to the level 0 which corresponds to processing of data without any testing for outliers/inliers. The computer program prints 2- to 1000-year floods from normal distributions after power transformation, both with and without kurtosis correction; from log-Pearson type III distributions, with sample skew and weighted skew; and from mixed distributions (Singh and Sinclair, 1972; Singh and Nakashima, 1981).

An outlier in a set of data is defined as an observation or a subset of observations, which appears to be inconsistent with the remainder of that set of data (Barnett and Lewis, 1978). The inconsistency can be interpreted as the observation being either significantly higher or lower at the high end (or lower or higher at the low end) than the value indicated by the rest of the data; the observation is termed as an outlier or inlier, respectively.

SUPPORTING INFORMATION

Statistical test for outliers and inliers

A literature search did not show the existence of statistical tests for checking outliers at higher than the 0.05 level or inliers at any level. The development of statistical tests for detection of outliers and inliers at various levels has been achieved (Singh and Nakashima, 1981) with experiments on millions of normally distributed random numbers generated with the polar method developed by Box, Jenkins, and Marsaglia

(Knuth, 1968). The derived statistics for an outlier at the high end at the 0.01 and 0.05 levels are the same as given by Barnett and Lewis (1978).

The developed test statistic is termed a departure and is given by the following expression:

$$\text{departure, } \Delta_i = z_i - z_{si} \tag{1}$$

in which z is the theoretical standard normal deviate and z_s is the sample standardized deviate. The departures were determined at 23 probabilities for sample sizes 10, 15, 20, 25, 30, 40, 50, 60, 75, and 100. Usually the number of outliers and inliers at the high and low end of the flood spectrum increases with the sample size. The distributions of departures for outliers and inliers for the 5 highest and 5 lowest floods (number 1 is the highest flood at the high end or the lowest flood at the low end) are graphed in figure 1. It is evident that the absolute value of the departure for the outlier is higher than for the inlier. The inlier at the high end cannot be too low; otherwise the next lower flood will replace it. This explains the difference between values of departures for outliers and inliers at corresponding probability levels.

Transformation of observed flood series

In order to use the test statistics for detection of outliers and inliers at various levels, the observed flood series first has to be transformed to resemble a series distributed as $N(\mu,\sigma^2)$ as closely as possible. The search for and testing of various transformations resulted in the selection of the power transformation (Box and Cox, 1964) as the best technique. This transformation follows the equations:

$$y_i = (Q_i^{\lambda} - 1)/\lambda \; ; \; \lambda \neq 0 \tag{2}$$

and

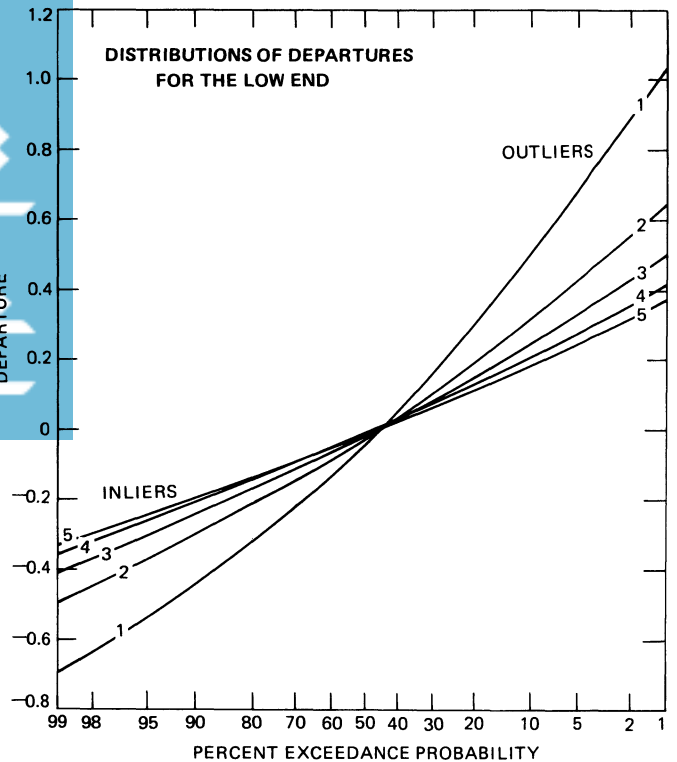
$$y_i = \log Q_i \qquad ; \; \lambda = 0 \tag{3}$$

in which Q is the annual flood from a sample of size n, and i = 1, 2, ..., n. It is a general power transformation, and the logarithmic, reciprocal, and square root transformations can be considered as its special cases. The constant λ can be obtained with the maximum log-likelihood method (Singh, 1980):

$$L_{max}(\lambda) = -\frac{1}{2} n \log \hat{\sigma}_y^2 (\lambda) + \log J(\lambda;Q) \tag{4}$$

and

$$\log J(\lambda;Q) = (\lambda - 1) \sum_{i=1}^{n} \log Q_i \tag{5}$$



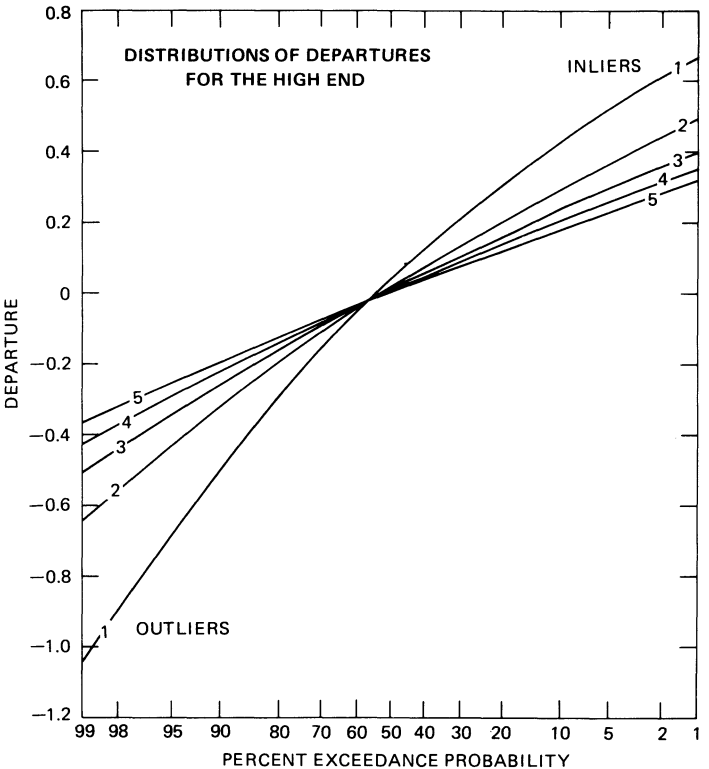


Figure 1. Distribution of departures for outliers and inliers

Kurtosis correction

The power-transformed series, y, has a skew very close to zero but the kurtosis, kt, may not equal 3 as for a normal distribution. The kurtosis correction factors were developed following the procedure outlined by Box and Tiao (1973). Values of standard deviates with kurtosis correction are given in table 1. Parameter β is related to kurtosis, kt, by the expression

$$kt = \frac{\Gamma[5(1+\beta)2]\ \Gamma[(1+\beta)/2]}{\{\Gamma[3(1+\beta)/2]\}^2} \tag{6}$$

Distributions used

Normal distribution The power-transformed series is considered a normal distribution, $N(x,s^2)$, in which x is the mean and s is the standard deviation of y series. The estimate for a T-year flood is obtained from

$$y_T = x + z_T s \tag{7}$$

in which z_T is with $\beta = 0$ without kurtosis correction or with β corresponding to kt for the y series. The y_T is then transformed to Q_T with inverse transformation

$$Q_T = (\lambda\ y_T + 1)^{1/\lambda} \tag{8}$$

Log-Pearson type III distribution or LP3 The power-transformed series is retransformed to the Q series after any detection and modification of outliers and inliers. The Q series is analyzed as an LP3 distribution (U.S. Water Resources Council, 1977) and the T-year flood estimate is obtained with the sample skew g_s as well as the weighted skew g_w. The weighted skew g_w is obtained from

$$g_w = g_s\ w + (1 - w)\ g_r \tag{9}$$

in which w equals (n-25)/75 and lies between 0 and 1, and g_r in the regional skew.

Mixed distribution The mixed distribution concept considers logarithms of annual floods to belong to 2 populations with means μ_1 and μ_2, variances σ_1^2 and σ_2^2, and relative weights α and $1-\alpha$. The mixed distribution method, developed from various studies (Singh and Sinclair, 1972; Singh and Nakashima, 1981), is based on the following equations:

$$p\ \{x\} = \alpha\ p_1\ \{x\} + (1 - \alpha)\ p_2\ \{x\} \tag{10}$$

$$p_1\ \{x\} = \frac{1}{\sigma_1\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{(x' - \mu_1)^2}{2\sigma_1^2}\right] dx' \tag{11}$$

Table 1. Values of z_T for Various Values of β and T

β	Values of z_T for Recurrence Interval, T, of					
	10	25	50	100	500	1000
-1.00	1.386	1.593	1.663	1.697	1.725	1.729
-0.95	1.384	1.592	1.665	1.708	1.762	1.777
-0.90	1.378	1.594	1.679	1.736	1.817	1.841
-0.85	1.372	1.600	1.699	1.769	1.875	1.908
-0.80	1.366	1.608	1.721	1.803	1.935	1.977
-0.75	1.360	1.618	1.744	1.839	1.996	2.047
-0.70	1.355	1.629	1.768	1.875	2.056	2.117
-0.65	1.350	1.640	1.792	1.911	2.117	2.187
-0.60	1.345	1.651	1.815	1.946	2.178	2.257
-0.55	1.340	1.661	1.838	1.982	2.238	2.328
-0.50	1.335	1.672	1.861	2.016	2.298	2.398
-0.45	1.330	1.682	1.883	2.050	2.358	2.468
-0.40	1.326	1.691	1.904	2.083	2.418	2.538
-0.35	1.321	1.700	1.925	2.116	2.477	2.608
-0.30	1.315	1.709	1.945	2.148	2.535	2.677
-0.25	1.310	1.717	1.965	2.179	2.594	2.747
-0.20	1.305	1.725	1.984	2.210	2.651	2.816
-0.15	1.299	1.732	2.002	2.240	2.709	2.885
-0.10	1.293	1.739	2.020	2.269	2.766	2.954
-0.05	1.288	1.745	2.037	2.298	2.822	3.022
0.00	1.282	1.751	2.054	2.326	2.878	3.090
0.05	1.275	1.756	2.070	2.354	2.934	3.158
0.10	1.269	1.761	2.085	2.381	2.989	3.226
0.15	1.263	1.765	2.100	2.407	3.044	3.293
0.20	1.256	1.770	2.114	2.433	3.098	3.361
0.25	1.249	1.773	2.128	2.458	3.152	3.428
0.30	1.243	1.776	2.141	2.482	3.205	3.494
0.35	1.236	1.779	3.154	2.506	3.258	3.561
0.40	1.229	1.782	2.166	2.529	3.311	3.627
0.45	1.222	1.784	2.178	2.552	3.363	3.692
0.50	1.214	1.786	2.189	2.574	3.414	3.758
0.55	1.207	1.787	2.200	2.596	3.465	3.823
0.60	1.200	1.788	2.210	2.617	3.516	3.888
0.65	1.192	1.789	2.220	2.637	3.566	3.952
0.70	1.185	1.789	2.229	2.657	3.616	4.016
0.75	1.177	1.790	2.238	2.677	3.665	4.080
0.80	1.169	1.789	2.247	2.695	3.714	4.143
0.85	1.162	1.789	2.255	2.714	3.762	4.206
0.90	1.154	1.788	2.262	2.732	3.810	4.269
0.95	1.146	1.787	2.269	2.749	3.857	4.331
1.00	1.138	1.786	2.276	3.766	3.904	4.393

$$p_2\{x\} = \frac{1}{\sigma_2\sqrt{2\pi}} \int_{-\alpha}^{x} \exp\left[-\frac{(x'-\mu_2)^2}{2\sigma_2^2}\right] dx' \qquad (12)$$

in which p is the probability of being equal to or less than x and x = log Q. Mixed distribution is a versatile distribution and can match most of the observed flood distribution shapes with proper values of α, μ_1, μ_2, σ_1, and σ_2. Kurtosis correction is valid only if the observed or power-transformed distribution is symmetrical. However, the mixed distribution allows for various combinations of skew, kurtosis, and asymmetries observed even after power transformation.

NEW FLOOD FREQUENCY METHODOLOGY

The concept of levels and windows is clarified in table 2 and figure 2. For the highest flood, the outlier H1 lies in window 1 if departure $\Delta \leq -1.054$, in window 2 if $-1.054 < \Delta \leq -0.683$, and so on for windows 3 through 6. If some outliers and/or inliers are detected in window 1, their departures are modified to respective values at level 1, and the procedure is followed sequentially from one window to the next. If no outliers and/or inliers are detected in a particular window, no modification is needed, and the program moves to the next window after developing and printing distribution statistics and flood estimates.

The flow chart

The detection and modification of outliers and inliers, as well as flood frequency analysis, follows the flow chart given in figure 3. Some relevant explanations to clarify the methodology and the computer program are given below. The sequence numbers correspond to the numbers attached to various boxes in the flow chart.

1.) Number of low as well as high floods, NO, can be provided as input information or computed from NO = [n/10] where NO = 5 for $n \geq 50$.

2.) Standard normal deviates for NO floods at both high and low end of the ranked flood series are obtained by converting p to z with a p-to-z subroutine, assuming normal distribution.

$$p = \frac{m - \alpha}{n + 1 - 2\alpha} \quad ; \; m = 1, 2, \ldots, n \qquad (13)$$

The value of α is obtained from the following information generated during development of departure test statistics.

Table 2. Test Values of Outlier and Inlier Departures

*	p	Outlier/ Inlier	Test values of departures: Low 1 15-100	Low 2 20-100	Low 3 25-100	Low 4 30-100	Low 5 40-100
1	<.01	Inlier	<-.689	<-.495	<-.412	<-.363	<-.327
	>.99	Outlier	>1.029	>0.643	>0.498	>0.418	>0.368
2	<.05	Inlier	<-.532	<-.369	<-.303	<-.264	<-.237
	>.95	Outlier	>0.681	>0.421	>0.337	>0.285	>0.253
3	<.10	Inlier	<-.441	<-.299	<-.243	<-.211	<-.188
	>.90	Outlier	>0.503	>0.321	>0.254	>0.217	>0.193
4	<.20	Inlier	<-.318	<-.209	<-.167	<-.143	<-.127
	>.80	Outlier	>0.297	>0.197	>0.159	>0.137	>0.123
5	<.30	Inlier	<-.221	<-.141	<-.110	<-.093	<-.082
	>.70	Outlier	>0.161	>0.112	>0.092	>0.081	>0.073
6	<.40	Inlier	<-.132	<-.080	<-.060	<-.050	<-.043
	>.60	Outlier	>0.052	>0.043	>0.037	>0.034	>0.032
			High 1 15-100	High 2 20-100	High 3 25-100	High 4 30-100	High 5 40-100
1	<.01	Outlier	<-1.054	<-.654	<-.511	<-.429	<-.377
	>.99	Inlier	>0.679	>0.488	>0.407	>0.358	>0.323
2	<.05	Outlier	<-.683	<-.433	<-.341	<-.290	<-.256
	>.95	Inlier	>0.529	>0.369	>0.300	>0.263	>0.235
3	<.10	Outlier	<-.500	<-.322	<-.258	<-.221	<-.195
	>.90	Inlier	>0.438	>0.299	>0.241	>0.209	>0.186
4	<.20	Outlier	<-.295	<-.197	<-.161	<-.139	<-.124
	>.80	Inlier	>0.317	>0.209	>0.166	>0.143	>0.126
5	<.30	Outlier	<-.159	<-.112	<-.094	<-.082	<-.074
	>.70	Inlier	>0.221	>0.140	>0.110	>0.093	>0.082
6	<.40	Outlier	<-.051	<-.043	<-.039	<-.035	<-.032
	>.60	Inlier	>0.132	>0.079	>0.060	>0.050	>0.043

Notes: 15-100, ..., and 40-100 denote the range of sample size n in years; * denotes window

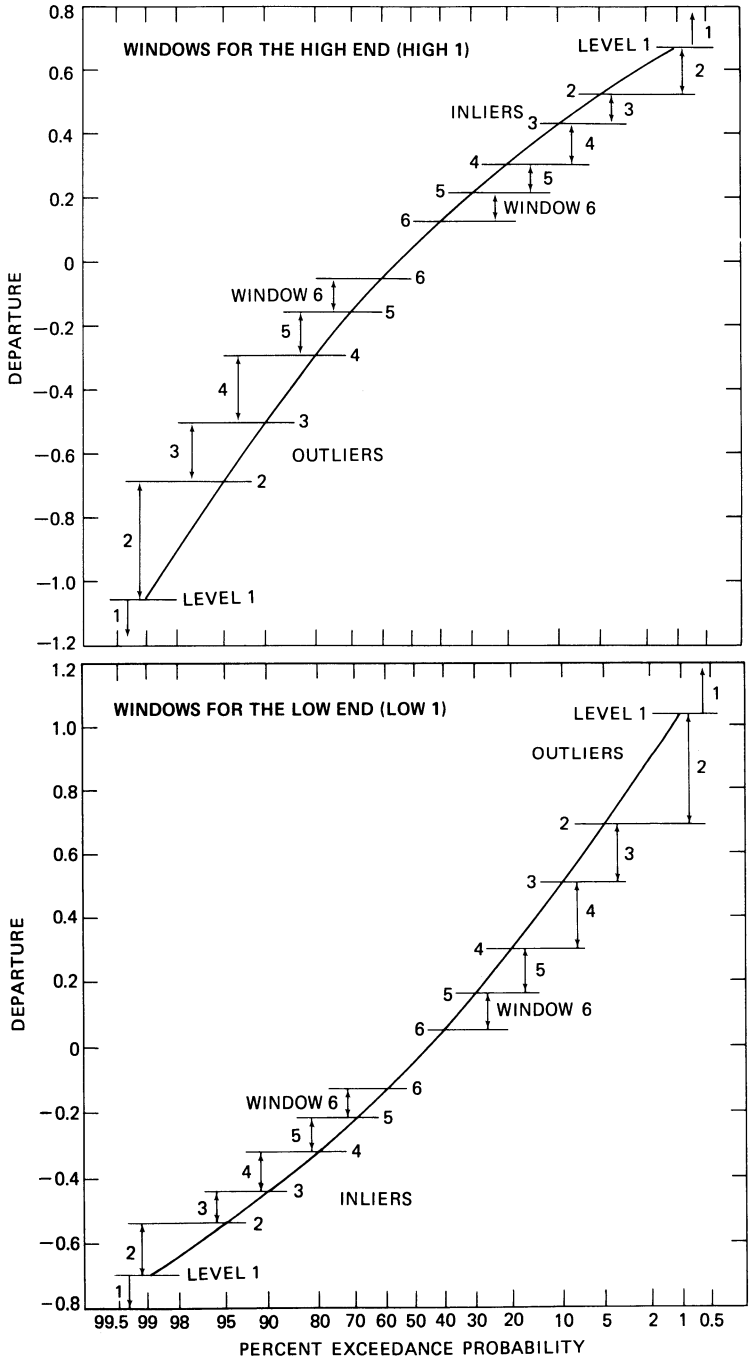


Figure 2. Levels and windows for outliers and inliers

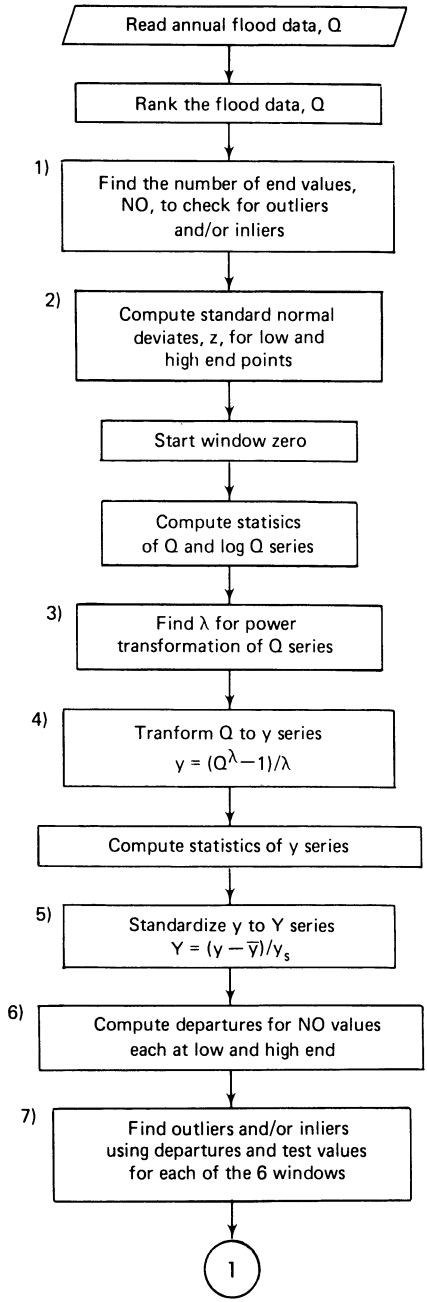


Figure 3. Flow chart for computer program

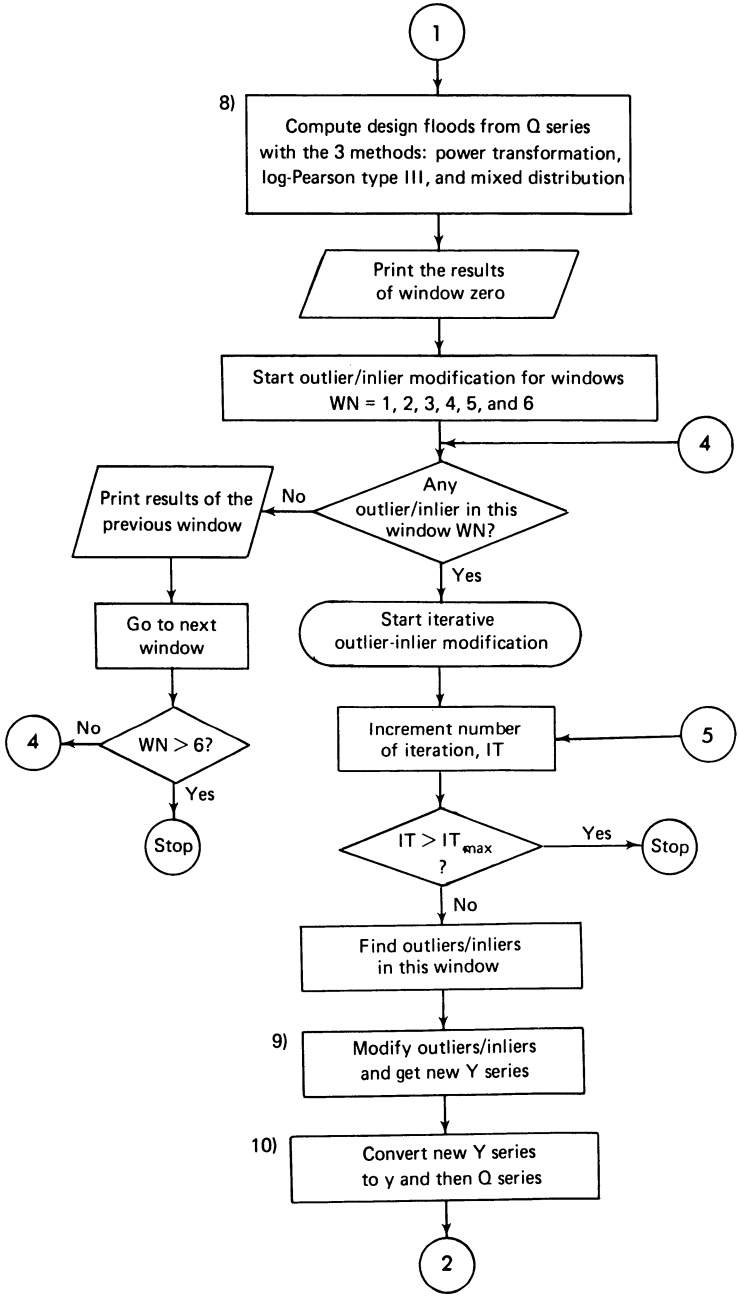


Figure 3. Continued

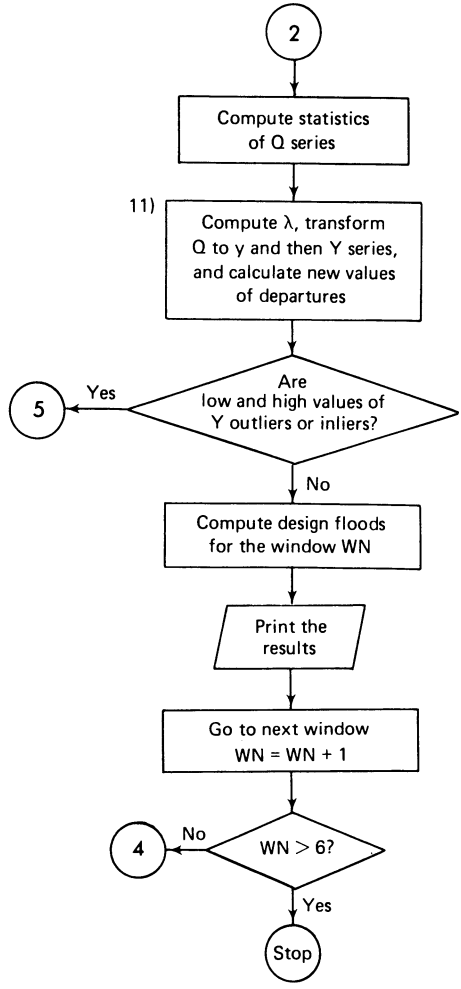


Figure 3. Concluded

n	values for the 5 highest and lowest ranks				
	1	2	3	4	5
10	0.425	0.474	0.492	0.506	0.511
15	0.414	0.464	0.485	0.498	0.506
20	0.408	0.455	0.478	0.491	0.501
25	0.406	0.448	0.472	0.486	0.496
30	0.404	0.443	0.467	0.481	0.491
40	0.403	0.440	0.459	0.473	0.482
50	0.403	0.440	0.454	0.467	0.475
60	0.403	0.440	0.451	0.462	0.469
75	0.403	0.440	0.450	0.458	0.463
100	0.403	0.440	0.450	0.456	0.460

3.) The parameter λ is computed with equations 2 and 3.

4.) The given Q series is transformed to a y series.

5.) The y series is standardized to a Y series with

$$Y_i = (y_i - \overline{y})/y_s \qquad (14)$$

in which $\overline{y}$ and y_s are the mean and standard deviation of the y series.

6.) The departures, Δ_m, for the NO values at the low end as well as at the high end are obtained from

$$\Delta_m = z_m - Y_m \qquad (15)$$

7.) Outliers and inliers, if any, are detected in each of the 6 windows according to the 6 levels, with the departure values taken from table 2.

8.) The floods corresponding to 2-, 10-, 25-, 50-, 100-, 500-, and 1000-year recurrence intervals are computed with the 3 distribution methods described earlier, without any modification of outliers and/or inliers, i.e., with window 0.

9-11.) The detection of outliers and inliers is initiated for window 1. Any detected outliers and inliers are modified to correspond to the threshold level. The new Y series is transformed to a y series and then to a Q series with the previous value of λ. A new λ is derived and the detection-modification process is repeated (usually 2 or 3 iterations) until no outliers and inliers are detected in the window under consideration. The final Q or y series is used in computing various T-year floods. The results are printed and the detection-modification process is applied to the next window.

An example

The methodology for flood frequency analysis with objective detection and modification of outliers and inliers is applied

to observed annual maximum floods for the Sangamon River near Oakley (drainage area 774 square miles and n = 27 years, NO = 2), Illinois, U.S.A. The flood series ranked in an ascending order of magnitude is 2390, 2660, 3020, 3020, 3120, 3140, 3240, 3900, 3950, 4150, 4220, 4500, 5070, 5500, 5850, 6000, 6150, 6190, 7400, 7840, 8350, 11700, 11800, 13200, 13700, 15300, and 16600 cfs.

The computer results for 100-year floods are given in table 3. The mixed-distribution parameters were determined with the generalized reduced gradient method and a nonlinear programming algorithm, the computer program for which was available from the University of Illinois. Two objective functions, i.e., minimization of $\Sigma(\Delta z)^2$ and $\Sigma|\Delta z|$, were considered, with 5 constraint equations (Cohen, 1967; Singh and Nakashima, 1981).

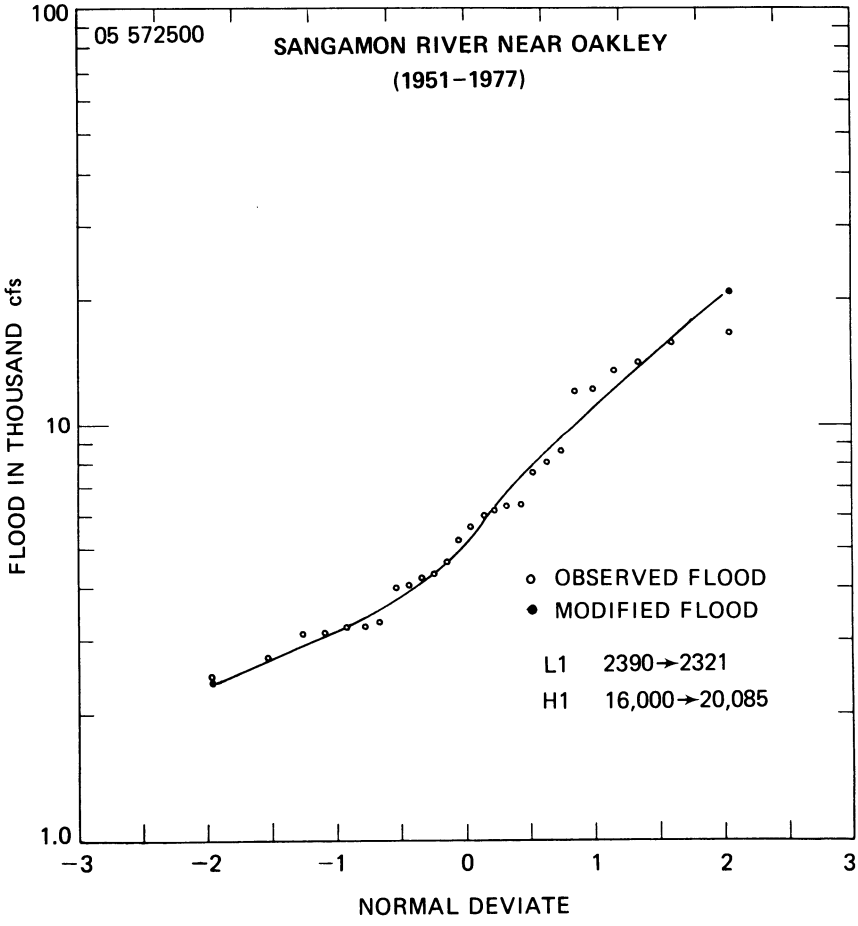


Figure 4. Observed and modified floods and the fitted mixed distribution curve (window 5)

Table 3. Flood Frequency Analyses: Sangamon River Near Oakley

STATION NO. 5572500 SANGAMON RIVER NEAR OAKLEY
DRAINAGE AREA 774.0 Sq Mi Years of Record 27 (1951-1977)

LEVEL NO.		0	1	2	3	4	5	6
METHOD		100-Year Flood in cfs						
Power Transform, PT								
With kt = 3.0		32,191	32,191	32,191	32,191	34,458	36,775	38,599
With sample kt		23,346	23,346	23,346	23,346	24,824	27,145	30,550
Log Transform								
LP3, Sample skew		25,630	25,630	25,630	25,630	26,685	28,136	30,069
LP3, Weighted skew		18,472	18,472	18,472	18,472	18,840	19,357	20,109
Mixed Distrib., MD		21,888	21,888	21,888	21,888	22,929	24,557	26,840
Type	No.	Observed and Modified Floods in cfs						
Low	1*	2,390	2,390	2,390	2,390	2,390	2,321	2,191
	2*	2,660	2,660	2,660	2,660	2,660	2,660	2,565
	3	3,020						
	4	3,020						
	5	3,120						
High	5	11,800						
	4	13,200						
	3	13,700						
	2*	15,300	15,300	15,300	15,300	15,300	15,300	15,619
	1*	16,000	16,000	16,000	16,000	17,766	20,085	22,841
METHOD	STATISTICS	Values of Statistics						
PT	mean	2.409	2.409	2.409	2.409	2.282	2.225	2.287
	std dev	.017	.017	.017	.017	.014	.013	.015
	skew	.093	.093	.093	.093	.098	.097	.086
	kurtosis,kt	2.202	2.202	2.202	2.202	2.232	2.311	2.455
	5th moment	.322	.322	.322	.322	.388	.403	.325
	lambda	-.402	-.402	-.402	-.402	-.427	-.439	-.426
LP3	mean	3.755	3.755	3.755	3.755	3.757	3.758	3.759
	std dev	.250	.250	.250	.250	.253	.258	.266
	sample skew	.398	.398	.398	.398	.439	.486	.528
	kurtosis,kt	2.309	2.309	2.309	2.309	2.395	2.551	2.773
	5th moment	1.997	1.997	1.997	1.997	2.373	2.934	3.612
MD	weight 'a'	.378	.378	.378	.378	.385	.362	.391
	mu1	3.523	3.523	3.523	3.523	3.531	3.536	3.558
	mu2	3.896	3.896	3.896	3.896	3.898	3.885	3.889
	sigma1	.089	.089	.089	.089	.095	.097	.120
	sigma2	.207	.207	.207	.207	.216	.235	.253
	Test Stat	2.955	2.955	2.955	2.955	2.654	2.331	2.132

* High & low floods considered for outlier detection and modification

Δz equals the difference between the standard deviate corresponding to p with $\alpha = 0.38$ (Blom, 1958; Cunnane, 1978) and that corresponding to p determined from equation 10. The asymmetry of the power-transformed Q series as in PT or log-transformed Q series as in LP3, as evidenced by the 5th moment being significantly different from zero, is accommodated easily by the mixed distribution concept.

The detection and modification of two inliers at the low end and two inliers at the high end from one level to the other are shown in table 3 together with relevant statistics for the 3 distribution methods. The test statistics under MD for indicating the goodness of fit is given by $\Sigma |\Delta z|$. The observed and modified floods and the fitted mixed distribution curve for window 5 are shown in figure 4.

REFERENCES

Barnett, V., and T. Lewis (1978) Outliers in Statistical Data. John Wiley and Sons, New York, 365 p.

Blom, G. (1958) Statistical Estimates and Transformed Beta Variables. John Wiley and Sons, New York, 176 p.

Box, G. E. P., and D. R. Cox (1964) An Analysis of Transformation. Journal of Royal Statistical Society, B26; 211-252.

Box, G. E. P., and G. C. Tiao (1973) Bayesian Inference in Statistical Analysis. Addison-Wesley, Reading, Mass., 588 p.

Cohen, A. C. (1967) Estimation in Mixtures of Two Normal Distributions. Technometrics, 9, 1:15-28.

Cunnane, J. (1978) Unbiased Plotting Position--A Review. Journal of Hydrology, 37, 205-222.

Knuth, D. E. (1969) The Art of Computer Programming. Volume 2. Addison-Wesley, Reading, Mass., 105-113.

Singh, K. P. (1980) Flood Frequency Analysis by Power Transformation--Discussion. J. Hyd. Div., ASCE, 106, HY3:462-465.

Singh, K. P., and R. A. Sinclair (1972) Two-Distribution Method for Flood Frequency Analysis. J. Hyd. Div., ASCE, 98, HY1: 29-44.

Singh, K. P., and M. Nakashima (1981) A New Methodology for Flood Frequency Analysis with Objective Detection and Modification of Outliers/Inliers. ISWS, Champaign, Illinois, 145 p.

U. S. Water Resources Council (1977) Guidelines for Determining Flood Flow Frequency. Bulletin 17A. Washington, D.C.

AN INTERACTIVE PACKAGE WITH GRAPHIC FACILITIES

A.Y. Barraud, Professeur à l'Inst. Nat. Polytech. de Grenoble
S. Gentil, Maître-Assistante à l'Inst. Nat. Polytech. de Grenoble
P. Laporte, Chercheur C.N.R.S.
C. Suleyman , Assistant à l'Inst. Nat. Polytech. de Grenoble

E.N.S. d'Ingénieurs Electriciens de Grenoble, Lab. d'Automatique,
B.P. 46, 38402 - Saint-Martin-d'Hères, France

INTRODUCTION

As computers are easily accessed and used today, new techniques have been developed allowing a very high-level dialogue between user and computer. Dialogue is well adapted to the modern teaching techniques and mini-computers available. When designing a control system, availability of interactive packages and graphical displays is of great interest. These facilities offer a large number of methods and sophisticated algorithms for control system design. At our department we wanted to have interactive tools to aid in undergraduate instruction and research in the area of Automatic Control theory.

I.S.E.R. - C.S.D. is meant to be a tool for solving problems in the field of : Simulation, Modeling, Identification, Parameter Estimation and Control of Dynamic Systems. The software environment with graphic facilities has been developed over a time period of almost ten years, emphasizing applications to computer aided undergraduate education of control engineering students as well as industrial processes control.

The experiences gained from utilization and maintenance of the system clearly indicate that it is very important that the program designer follows well defined design and documentation guidlines. Choice of the programming language is also of crucial importance for the sake of portability, structured programming and team work.

PASCAL is used as programming language for the main parts of the system and FORTRAN language is used for the codification of the application programs.

COMPUTER FACILITIES AND SYSTEM ORGANIZATION.

The program resides on Nord Computers operating under SINTRAN III/

VS. Its time sharing permits multiuser and multiprogramming facilities. Graphic displays are used for data inputs and outputs. ISER-CSD interpretes a high-level language for Automatic Control. This communication language is based on a very simple syntax and sufficiently flexible to meet demands from the experienced user as well as the casual user and fresh beginner. Reserved words are those used by specialistsin Automatic Control (words like SYSTEM, DATA, INPUT, OUTPUT, STATE, SIGNAL, STEP, etc...). Language has a menu structure to allow an easy manipulation of mathematical models which can be described in continuous or discrete forms.

All informations about an application are memorized on a DATA-STRUCTURE which is organized as an "n'airy" tree. As Pascal language well adopted for data structuring we use structures like records and pointers in organizing data. The DATA-STRUCTURE is dynamically managed facilitating systems association, reduction and stepwise evaluation. The DATA-STRUCTURE gives several versions or states of a problem during an application.

Graphics and the advantages offered in data input/output are very important for rapid problem analysis and synthesis with respect to input conditions and results obtained. An important effort has been done for the realization of a module as sophisticated as possible for graphical communications with the computer.

SOFTWARE CAPABILITIES

The program packages in ISER-CSD are designed to cover a wide range of control engineering problems, not only analysis and design of control systems but also system identification, parameter estimation and simulation of large scale and complex systems. Different model representation forms are handled. For instance : multivariable state space models in continuous or discrete form, multivariable transfer matrix in continuous or discrete form or matrix polynomial forms. Transformations between the different model descriptions are performed and equivalent systems are calculated. Time-domain models or frequency domain-models can be designed using physical approach called "Modeling" where the process model is described by means of a set of differential and algebraic equations. Alternatively one can derive a model by means of experiments on the real process followed by a system identification. Specialized programs can help for data analysis and manipulation. Main applications of the ISER-CSD system are :

Time Domain Applications.

. Simulation,
. Equivalent-systems computation,
. Stability test,
. Controllability / Observability test,
. Determination of noncontrollable and/or nonobservable modes,

. Pole placement,
. Closed-loop systems simulation,
. Computation of Kalman-Bucy filters,
. Model following,
. Resolution of the optimal regulator problems,
. System identification and studying systems under influence of noise.

Frequency Domain Applications.

. Stability analysis,
. Transfer matrix calculation,
. Transient and stationary behaviour analysis of systems,
. Bode, Nyquist and Nichols plots of frequency characteristics,
. System reduction and closed-loop system analysis.

Industrial and Advance Applications.

. Hierarchical control design,
. Simulation of nonlinear systems,
. Modelling, (using theory of BOND-GRAPHS),
. Graphical input/output of dynamic systems,
. Aircraft simulation,
. Studying large-scale systems.

IDENTIFICATION TECHNIQUES

ISER-CSD supports a relatively large number of identification algorithms which mainly belong to two classes of methods :"off-line" or iterative approaches based upon non-linear optimization routines, and "on line" techniques embodied as a set of recursive algorithms implementing several filter schemes.

Iterative Methods.

Two methods are available through ISER-CSD : the so-called "output error method", see figure 1, and the maximum likelihood method, figure 2, which operate upon MISO systems structure (Multiple Input - Single Output).

These two figures exhibit some notations which are : $\underline{u}_t$ the input vector of dimension m, the measured output y_t, η_t the model output, e_t the output error, $B(z^{-1})$ a polynomial with m-vector coefficients $\underline{b}_i$, $A(z^{-1})$ another polynomial which must be read : $A(z^{-1}) = 1 - (a_1 z^{-1} + \ldots\ldots + a_n z^{-n})$, a filter $C(z^{-1})$ and lastly ε_t the innovation sequence. z^{-1} has the classical meaning of shift-operator (delay) : $z^{-1}\eta_t = \eta_{t-1}$.

Both methods search for a minimum of a non linear criterion via iterative algorithms.

Output error method. Let us consider the following n order model following from figure 1 :

$$\eta_t = a_1\eta_{t-1} + \ldots + a_n\eta_{t-n} + \underline{b}_o^T \underline{u}_t + \ldots + \underline{b}_n^T \underline{u}_{t-n} \qquad (1)$$

Then given a set of input/output data $\{\underline{u}_t , y_t\}$ $t = 1, 2, \ldots, N$; the optimal parameters model $\hat{\theta}$ ($\underline{\theta} = \{a_1, \ldots, a_n, \underline{b}_o, \ldots, \underline{b}_n\}$) are solution of the optimization problem :

$$\min_{\underline{\theta}} J(\underline{\theta}) = \sum_{t=1}^{N} [\eta_t(\underline{\theta}) - y_t]^2 \qquad (2)$$

which has been numerically solved by the Powell's quasi-Newton algorithm VAI3A, Powell, 1975. This routine requires the explicit knowledge of the criterion gradient :

$$\frac{\partial J}{\partial \underline{\theta}} = 2 \sum_{t=1}^{N} (\eta_t - y_t) \frac{\partial \eta_t}{\partial \underline{\theta}} \qquad (3)$$

which is computed through the well-known sensitivity equations :

$$\begin{cases} \dfrac{\partial \eta_t}{\partial a_i} = \eta_{t-i} + \sum_{j=1}^{n} a_j \dfrac{\partial \eta_{t-j}}{\partial a_i} \\ \dfrac{\partial \eta_t}{\partial \underline{b}_i} = \underline{u}_{t-i} + \sum_{j=1}^{n} a_j \dfrac{\partial \eta_{t-j}}{\partial \underline{b}_i} \end{cases} \qquad (4)$$

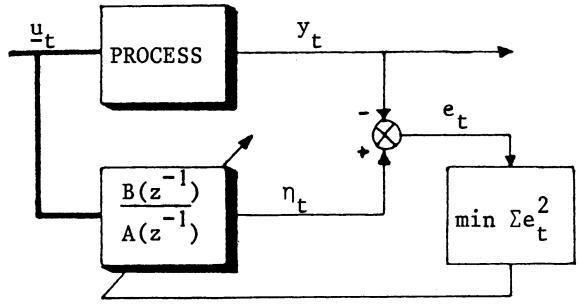
It must be pointed out that the criterion (Equation 2) is a quadratic function of the $\underline{b}_i$ and a very non linear one of the a_i. When m is large it can be worthwhile to work with a modified criterion in a reduced space of dimension only n ($\underline{\theta}$ beeing of dimension n(m+2)) obtained by projection :

$$\min_{\underline{\theta}} J(\underline{\theta}) = \min_{a_i} [\min_{\underline{b}_j} J(\underline{\theta})] = \min_{a_i} J^*(a_i) \qquad (5)$$

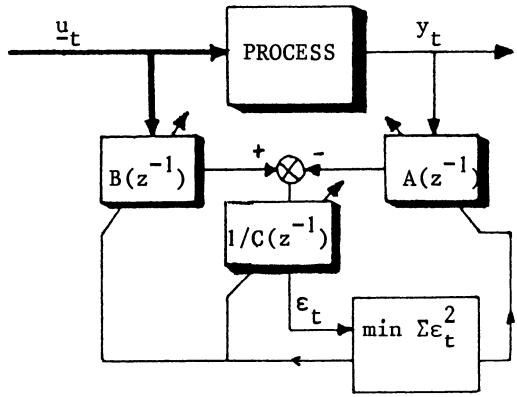
where J^* is explicitly computed as the optimal value of a classical linear least squares problem for each given values of a_i. Then the iterative optimization routine operates upon J^* whose gradient is easily obtained as :

$$\left. \frac{\partial J(\underline{\theta})}{\partial a_i} \right|_{\underline{b}_i = \hat{\underline{b}}_i} \qquad (6)$$

i.e. the value deduced from equation (4) with $\hat{\underline{b}}_i$ the value of $\underline{b}_i$ minimizing $J(\underline{\theta})$ and leading to $J^*(a_i)$, by definition.



Output error method — Figure 1.



Maximum likelihood method — Figure 2.

Maximum likelihood method. According to figure 2, the model must now be read :

$$y_t = a_1 y_{t-1} + \ldots + a_n y_{t-n} + \underline{b}_o^T \underline{u}_{t-1} + \ldots + \underline{b}_n^T \underline{u}_{t-n} + \varepsilon_t + c_1 \varepsilon_{t-1} + \ldots + c_n \varepsilon_{t-n} \quad (7)$$

with the larger set of parameters $\underline{\theta} = \{a_1, \ldots, a_n, \underline{b}_o, \ldots \underline{b}_n, c_1, \ldots, c_n\}$ whose optimal value $\hat{\underline{\theta}}$ is solution of the new optimization problem :

$$\min_{\underline{\theta}} J(\underline{\theta}) = \sum_{t=1}^{N} \varepsilon_t^2 \quad (8)$$

This is a kind of "equation-error" method.

As it can be seen, ε_t is linear in a_i and $\underline{b}_i$ but quite non-linear versus c_i, in a similar way to the a_i in the output error criterion (Equation 2). Well known expression have been derived

for computing efficiently the gradient and Hessian of problem (Equation 8). Such an implementation has been done in AUTOPACK, Barraud, 1979 using Hebden's Newton algorithm, Hebden, 1973, to solve equation (8).

For both approaches, the user may specify an initial value $\underline{\theta}_o$ to start the iterative scheme, or asks for a self starting procedure which, here, consists in a classical linear least square step. This common quadratic optimization problem is respectively obtained by putting y_t in place of η_t in equation (1) and setting the c_i to zero in equation (7).

Remark. The user may indicate that some of the $\underline{b}_i$ and/or the c_i are zero in order to vary the complexity of the model for a given order n.

Recursive Methods.

There are six basic schemes available under ISER-CSD to run a recursive identification : LS (Least Squares), GLS (Generalized Least Squares), YIV (Young Instrumental Variables), LAN (Landau Hyperstable parallele model), BAN (Banon Instrumental variables with delayed observations), IVP (Instrumental Variables with delayed Parameters). All these methods have been set up in a same framework by Bethoux, 1976 :

$$\tilde{\varepsilon}_{t+1} = y_{t+1} - \hat{\underline{\theta}}_t^T \underline{x}_t + \underline{c}^T \underline{e}_t \quad ; \quad \hat{\underline{\theta}}_{t+1} = \hat{\underline{\theta}}_t + P_t \underline{Z}_t \tilde{\varepsilon}_{t+1}$$

$$P_{t+1} = P_t - P_t \underline{Z}_t (1 + \underline{x}_t^T P_t \underline{Z}_t)^{-1} \underline{x}_t^T P_t \qquad (9)$$

with $\underline{x}_t$, $\underline{Z}_t$, $\underline{c}$ and $\underline{e}_t$ defined as follow :

Method	$\underline{x}_t^T$	$\underline{Z}_t^T$	$\underline{c}^T$	$\underline{\theta}^T$
L.S.	$\underline{v}_t^T$	$\underline{v}_t^T$	[0,..,0]	$[a_1,\ldots,a_n:\underline{b}_1,\ldots,\underline{b}_n]$
G.L.S.	$[\underline{v}_t^T,\underline{e}_t^T]$	$[\underline{v}_t^T,\underline{e}_t^T]$	[0,..,0]	$[a_1,..,a_n:\underline{b}_1,\ldots,\underline{b}_n:c_1,\ldots,c_n]$
V.I.V.	$\underline{v}_t^T$	$\underline{Za}_t^T$	[0,..,0]	$[a_1,..,a_n:\underline{b}_1,\ldots,\underline{b}_n]$
L A N	$\underline{Z\ell}_t^T$	$\underline{Z\ell}_t^T$	$[c_1,..,c_n]$	$[a_1,..,a_n:\underline{b}_1,\ldots,\underline{b}_n]$
B A N	$\underline{v}_t^T$	$\underline{Zo}_t^T$	[0,..,0]	$[a_1,..,a_n:\underline{b}_1,\ldots,\underline{b}_n]$
I V P	$\underline{v}_t^T$	$\underline{Zr}_t^T$	[0,..,0]	$[a_1,..,a_n:\underline{b}_1\ldots,\underline{b}_n]$

where the notations are :

$$\underline{v}_t^T = [y_t,y_{t-1},\ldots,y_{t-n+1}:\underline{u}_t,\underline{u}_{t-1},\ldots,\underline{u}_{t-n+1}] \qquad (10)$$

$$
\begin{cases}
\underline{e}_t^T = [e_t, e_{t-1}, \ldots, \underline{e}_{t-n+1}] \; ; \; e_t = y_t - (\sum_{i=1}^{n} a_i y_{t-i} + \sum_{i=1}^{n} \underline{b}_i^T \underline{u}_{t-i}) \\
\underline{Za}_t^T = [Za_t, .., Za_{t-n+1} : \underline{u}_t, \underline{u}_{t-1} .., \underline{u}_{t-n+1}] \; ; \; \underline{Za}_{t+1} = \hat{\underline{\theta}}_{t+1}^T \underline{Za}_t \\
\underline{Z\ell}_t^T = [Z\ell_t, .., Z\ell_{t-n+1} : \underline{u}_t, \underline{u}_{t-1}, \ldots, \underline{u}_{t-n+1}] \; ; \; Z\ell_{t+1} = \hat{\underline{\theta}}_t^T . \underline{Z\ell}_t \\
e_t = y_t - Z\ell_t \\
\underline{Zo}_t^T = [y_{t-r}, .., y_{t-n+1-r} : \underline{u}_t, .., \underline{u}_{t-n+1}] \\
r \text{ delay parameter} \\
\underline{Zr}_t^T = [Zr_t, .., Zr_{t-n+1} : \underline{u}_t, .., \underline{u}_{t-n+1}] \; ; \; Zr_{t+1} = \hat{\underline{\theta}}_{t-r}^T \underline{Zr}_t
\end{cases}
\tag{10}
$$

From a numerical point of view, a very important feature is that when $\underline{Z}$ and $\underline{x}$ are identical, a very nice implementation can be done using the information square root filter structure. Recall that a scheme such as :

$$
\begin{aligned}
P_{t+1} &= P_t - P_t \, \underline{a}_{t+1} (1 + \underline{a}_{t+1}^T P_t \, \underline{a}_{t+1})^{-1} \underline{a}_{t+1}^T P_t \\
\hat{\underline{\theta}}_{t+1} &= \hat{\underline{\theta}}_t + P_{t+1} \, \underline{a}_{t+1} (y_t - \underline{a}_{t+1}^T \hat{\underline{\theta}}_t) \; ; \; y_t = \underline{a}_t^T \underline{\theta} + \varepsilon_t
\end{aligned}
\tag{11}
$$

can be firstly rewritten for P_t as :

$$
P_{t+1}^{-1} = P_t + \underline{a}_{t+1} \, \underline{a}_{t+1}^T \tag{12}
$$

and secondly transformed into :

$$
Q \begin{bmatrix} L_t & \underline{d}_t \\ \underline{a}_{t+1} & y_{t+1} \end{bmatrix} = \begin{bmatrix} L_{t+1} & \underline{d}_{t+1} \\ 0 & 0 \end{bmatrix} \tag{13}
$$

where Q is an orthogonal matrix (givens rotations in our implementation, Barraud, 1980) computed such as the last row of the data matrix to be transformed, is set to zero. Then it can be easily verified that :

$$
\hat{\underline{\theta}}_{t+1} = (L_{t+1})^{-1} \, \underline{d}_{t+1} \; ; \; P_{t+1} = (L_{t+1}^T \, L_{t+1})^{-1} \tag{14}
$$

with initial conditions $L_o = 0$ and $\underline{d}_o = 0$. It must be noticed that $\hat{\underline{\theta}}_{t+1}$ does not depend on $\hat{\underline{\theta}}_t$ and then can only be computed as necessary, clearly the same remark applies to the covariance matrix P_t. Lastly parameter tracking can be easily embodied with the exponential decay technique which corresponds to a modified criterion :

$$
J(\underline{\theta}) = \sum_{t=1}^{N} (y_t - \underline{a}_t^T \underline{\theta})^2 (\lambda^2)^{N-t} \; ; \quad 0 < \lambda < 1 \tag{15}
$$

In such a case the data matrix introduced in the left hand side of equation (13) is multiplied by λ before introducing and

zeroing the last row $[\underline{a}_{t+1}^T, y_{t+1}]$.

APPLICATION TO AN UNDERGROUND WATER LEVEL MODEL

The System.

The system under study is a 70 km^2 swampy area located near the Rhône river (France). A part of the year, this region is flooded; this implies a mosquitoes eggs hatching as larvae need water to develop ; this has been for a long time a pest for cattle and human beings. Nowadays an antimosquito organization has been created, but this force intends to work in a way which is not prejudicial to the environment and so avoids an intensive use of chemical products : they periodically measure the water levels in nine piezometers and scatter the products only when the swamps are begining to be flushed.

Two reasons motivated our modelling study : first, getting a better insight in the water level fluctuations, and particularly, determining if the dominant reason for increasing level is rainfall or infiltration from the riverbed ; secondly, testing the possibility of water level prediction, which could avoid tedious field measurements and help to prepare the chemical scattering material.

The inputs considered for the ground water level system are : the air temperature, the precipitations and the river flow. The National Meteorological station of Amberieu supplies minimal and maximal temperatures ($^{\circ}$C) and the rainfall values (mm). The Compagnie Nationale du Rhône supplies the river flow (daily mean in m^3 /s). Weekly means are used for temperature and river flow,F, and a weekly integral is used for rainfall, R, as level measurements in the piezometers, L, are made weekly.

The study is based on an eight years data collection.

The Model.

Conceptual models of ground-water developed on physical arguments give satisfactory explanation of hydrological phenomena, but due to the large number of unidentifiable parameters such models introduce, they are overparametrized for estimation and prediction purposes, Fjeld and Aam, 1981. Linear models of the form studied in the previous paragraph seem better adapted to a practical field study.

In a linear model, the soil is seen as a series of tanks, the first being supplied by rainfall and submitted to evapotranspiration. The last tank, representing the aquifer, is also connected to the river, represented by another tank at an intermediate level. All these concepts are represented figure 3. So we look for the parameters of a transfer-function relating the water level

(output) to the three inputs. The evapotranspiration has been shown proportionnal to the maximal temperature T, Laporte, 1982. The representation is a discrete one, with a sampling period of one week corresponding exactly to the measurement period. The general mathematical formulation of the model is :

$$L(k) = \sum_{i=1}^{n} a_i L(k-i) + \sum_{j=1}^{p_1} b_{j,1} F(k-j) + \sum_{j=1}^{p_2} b_{j,2} T(k-j) + \sum_{j=1}^{p_3} b_{j,3} R(k-j) \qquad (16)$$

This representation is equivalent to the transfer-function one used in equation (1) and all the identification package has been tested on the underground-water system data set.

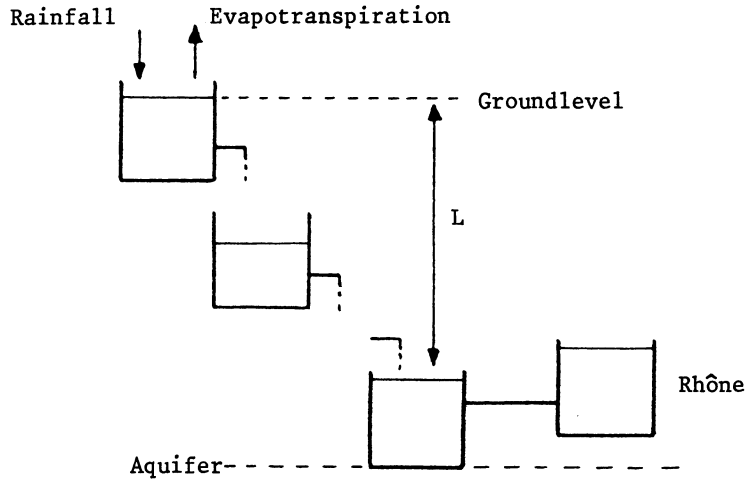


Figure 3. Tank model of soil.

The Results.

Tests were undertaken to determine the model order n and the numbers p_1, p_2, p_3, in equation (16). The choice of n must be based on several factors and needs some experience, Foulard, Gentil, Sandraz, 1976. The criterion J as defined in equation (2 or 8) diminishes generally when increasing n ; but for values of $n > n_o$, one observes that the decrements are much smaller than for $n < n_o$, so n_o is a good choice for n. The examination of the transfer-function poles (the zeros of the polynomial $A(z^{-1})$ are computed by each program) is also fruitfull : when n

is too large, there appear either near-zero poles, either negative ones ; the first kind have a negligible contribution to the output, the second one an oscillating contribution which tries to describe the measurement noise. The programs for output error or equation error methods compute the diagonal of the inverse of the Hessian of the criterion at the optimum, which can be shown to represent the parameters variance. This is useful to decide if a parameter has a statistical meaning or is to be set to zero. This helps to choose the n and the p_i of the models.

As an example of this work, the figure (4) reports the results of the order selection in the piezometer n°2 : it is not easy to take a decision in view of the models outputs, so the tests mentionned in the previous section were used to decide that $n=p_1=p_2=p_3=1$. The model is the simpler possible. An interesting feature is that it is true for all the piezometers, indicating an homogeneity of the soil behaviour in all this region, at least from the infiltration point of view.

Another interesting question upon this system is : have all the inputs an equal importance ? People involved in the field measurements had an empiral piezometers classification as follows : "influenced by the rain" or "influenced by the river".

The identification results of equation (16) show that this classification corresponds in fact to systems with high gains with respect to the three inputs and systems with low gains. But none input appears to have a negligible gain compared to the other inputs. This could be the result of a high correlation between them : the rainfall and the river-flow are evidently related ; the temperature and the river-flow are also related in summer through snow-melting. To solve this problem, identification was undertaken with one input at a time. It was shown that some piezometers classified as "influenced by the river" can effectively be described using only the river flow as input (see figure 5 as an example) but that others, classified as "influenced by the rain" cannot be correctly described without the river flow in some seasons (figure 6). So the models help to detect some seasonnal features which were not perceived by the global empirical analysis.

The predictive value of the models is illustrated by figure 5 : the parameters identified using the 1972 data set are used to predict the 1974-75 behaviour ; the inputs used for this prediction are the actual measurements of the rainfall, the temperature and the river flow.

The model may also be used for short-term forecasting using measurements of the water level and the inputs at week k to forecast the water level at week $k+1$. Recursive algorithms were used to track parameters variations. In fact (figure 7), one observes that the parameters are almost constant all the year : no seasonnal phenomenons could be detected.

CONCLUSION.

The interactive package with graphic facilities developed at ENSIEG has been rapidly described. The emphasis has been put on the Identification programs compatible with it. The application shows clearly the relevance of estimation theory to practical water resources problems. Even in the case of this complex system, generally described by partial differential equations, the study of linear models has been very fruitfull. We have observed a good homogeneity between each piezometers ; we have determined which inputs are predominant in each places. The predictive capacity of the model is good and we are now envisaging on-line implementation of the algorithms as a forecasting aid to people involved in field measurements. All this work has been realized in a very short time thanks to the availability and the versatility of the general purpose identification package.

REFERENCES.

Armstrong E.S. (1981) Interactive Systems for Education and Research in Control System Design, IEEE, International Conference on Cybernetics and Society, ATLANTA, Oct.

Barraud, A.Y. (1980) Identification des Invariants Fondamentaux, Approche Numérique. RCP 567 "Outils et Méthodes Mathématiques pour l'Automatique, l'Analyse des Systèmes et le Traitement du Signal" Vol. 1, p. 147-172.

Barraud, A.Y. (1979) AUTOPACK - A Numerical Software in Automatic Control. International Conference on Information Sciences and Systems, July 9-13, Patras, Greece.

Béthoux, G. (1976) Approche Unitaire des Méthodes d'Identification et de Commande Adaptative des Procédés Dynamiques. Thèse de Doctorat 3° Cycle, I.N.P. Grenoble.

Fjeld, M., Aam, S. (1981) An Implementation of Estimation Techniques to a Hydrological Model for Prediction of Runoff to a Hydroelectric Power Station. Modeling, Identification and Control, Vol. 2, n°1, 37-56.

Foulard, C., Gentil, S., Sandraz, J.P. (1976) Commande et Régulation par Calculateur Numérique : de la Théorie aux Applications Eyrolles Ed.

Hebden, M.D. (1973) An Algorithm for Minimization Using Exact Second Derivatives. T.P. 515, U.K.A.E.A., Harwell, England. March.

Laporte, P. (1982) Modélisation de la Nappe Phréatique dans le Bassin Versant du Haut-Rhône. Internal Report, Lab. d'Automatique de Grenoble.

Powell, H.J.D. (1975) Some Global Convergence Properties of a Variable Metric Algorithm for Minimization without Exact Line Searches. A.M.S./SIAM Symposium on Nonlinear Programming.

Suleyman, C. (1981) Interactive Systems for Education and Research in Control System Design. IEEE, International Conference on Cybernetics and Society, Atlanta, USA, Oct.

Suleyman, C. (1981) An Interactive Computer Package for Simulation and Design of Control Systems. Arab School on Science and Technology, Bloudan-Syria.

Suleyman, C. (1981) Conception et Réalisation d'un Système d'Aide à l'Enseignement et à la Recherche en Automatique. Thèse de Doctorat de 3° Cycle, I.N.P. Grenoble.

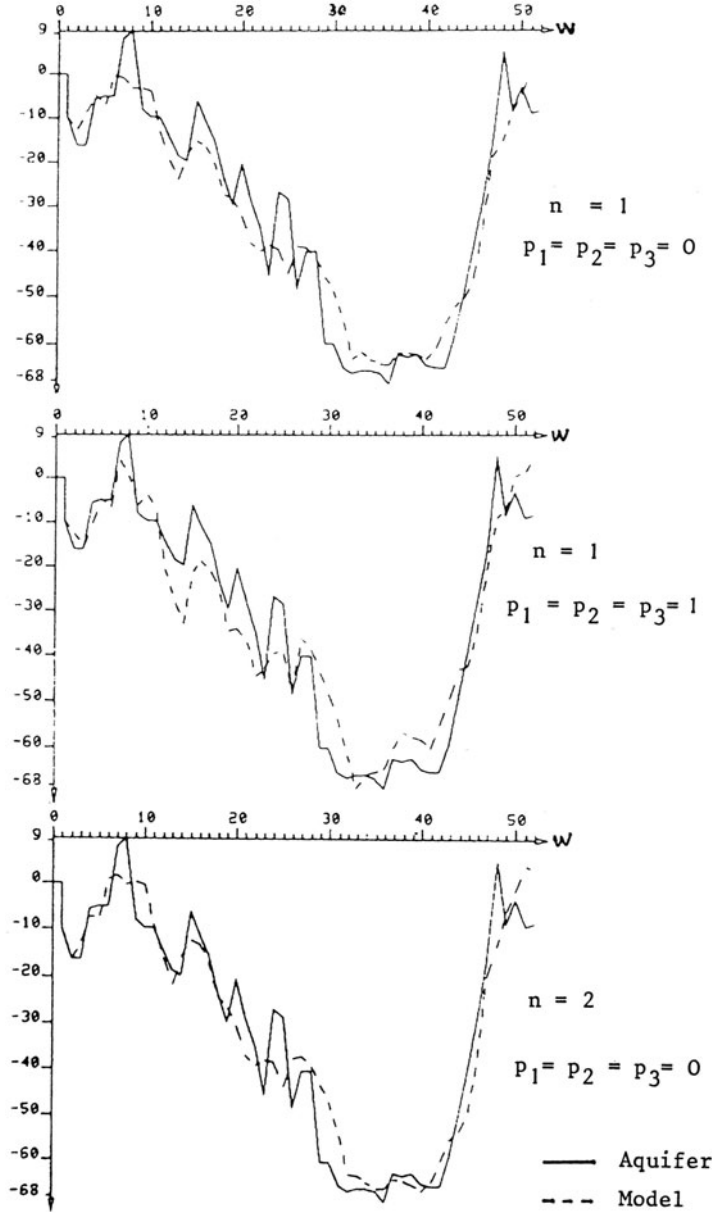


Figure 4. Determination of the model structure.

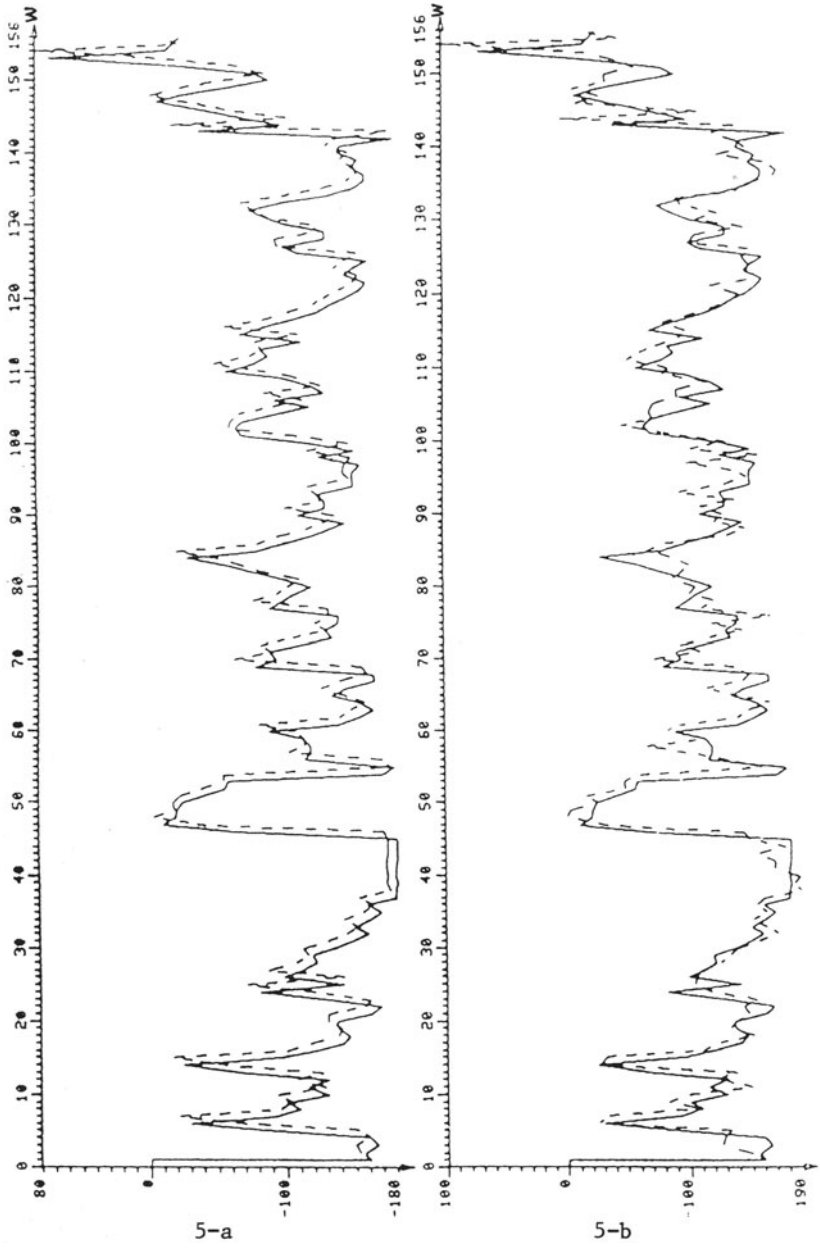
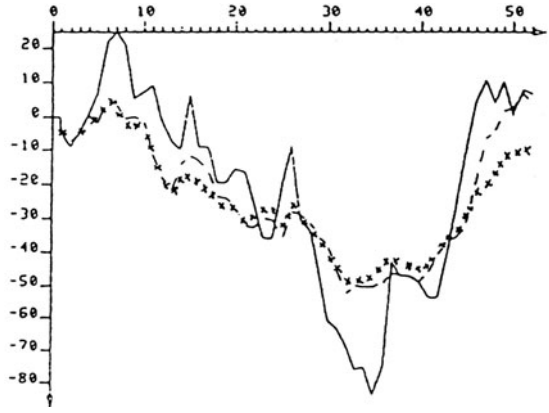


Figure 5. Predictive output of the model (---)

5-a 1 input : river flow
5-b 3 inputs : temperature, rainfall and riverflow.



1 : ——— Aquifer level 2 : - - - 3 inputs: temperature, rainfall and riverflow

3 : xxxxxx 2 inputs : temperature, rainfall

Figure 6. Influence of the riverflow.

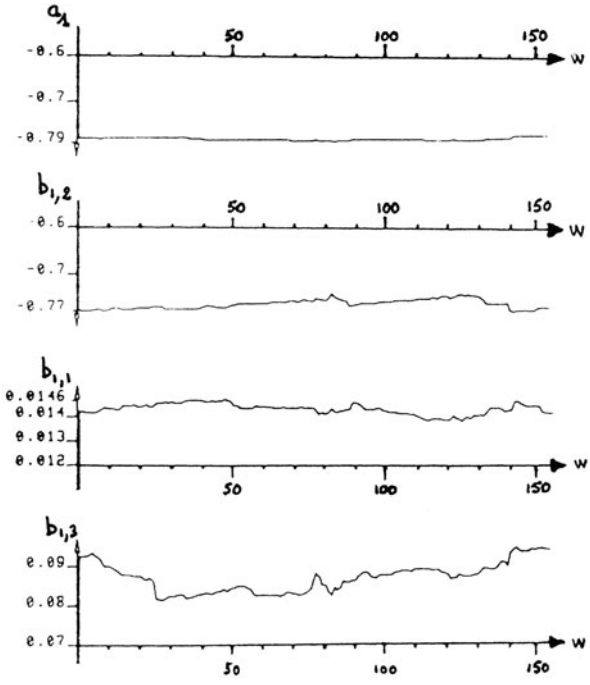


Figure 7. Recursive Identification : parameters variations from 1972 to 1974.

MICRO COMPUTER CONTROLLED LABORATORY TESTS ON A STORM WATER OVERFLOW CHAMBER

E.A. Delo
Dr. A.J. Saul
Simon Engineering Laboratories,
University of Manchester,
Manchester M13 9PL

ABSTRACT

The paper describes the development of a laboratory system in which the hydraulic performance of a double high side-weir storm water overflow chamber was tested under storm flood conditions. The unsteady discharge/time hydrograph was generated by a butterfly valve controlled by a microcomputer and the hydraulic performance of the chamber was monitored and recorded by a water level transmitter located at the downstream end of the overflow weirs.

A brief description of the overall concept of the system's operation is followed by an outline of the hardware associated with the control of the valve and the principles governing its real time operation. Hardware components used include a CBM 8032 micro with printer, disk drives and plotter, Pupi interface, E/P transducer, air operated butterfly valve, rotary potentiometer and water level transmitter.

The paper also outlines the structure of the five major software programs which were developed to perform the following functions: calibrate the system; describe the storm profile; control of valve, analysis, display, printing and storage results; graphical output of data; and maintain a data base of test runs. The procedure for transfer of data between the program is also detailed.

The final part of the paper gives the results from a test run and shows how the solution of an engineering problem has been achieved successfully with the aid of a microcomputer.

INTRODUCTION

A storm water overflow is a structure used on combined sewerage systems in which, both foul and surface water are transported in the same pipe. The purpose of the overflow is to remove excess water from the system in times of storm and to limit the pollutant load in that water discharged to the receiving watercourse.

In simple terms the incoming sewer pipe discharges into a chamber which has two outlets. The first is a much smaller sewer pipe/penstock arrangement at the downstream end of the chamber which controls the flow forwarded to treatment, while the second is a weir/siphon arrangement which may be positioned on the sides or end wall of the chamber. In dry conditions the foul sewage which constitutes the dry weather flow (dwf) is easily discharged through the chamber to treatment. However, in storm conditions the flow in the inlet pipe is increased significantly as a result of run off from roads, roofs and impermeable areas and indeed the flow entering the chamber may well exceed 50 dwf. As the outlet control at the downstream end of the chamber is normally designed to discharge 3-12 dwf to treatment at peak design inflow it is a frequent occurrence that the water depth in the chamber is increased to a level sufficient to cause overflow to the watercourse. The overflow continues to operate until the inflow has receded to a value which may be satisfactorily discharged by the outlet to the treatment works.

During operation the water discharged to the watercourse is likely to contain some crude sewage and other pollutant load. The level of this pollution at any moment in time is dependent on the nature of the storm, the characteristics of the sewage, the geometric configuration of the chamber and the antecedent conditions in the sewer prior to the storm. To reduce this watercourse pollutant load the geometric configuration of the chamber should be so chosen as to maximise the pollutant load discharged to treatment. The use of hydraulic scale models to facilitate the optimisation of chamber design to achieve this end has been adopted by a number of previous researchers. Sharpe and Kirkbride, 1959, Frederick and Markland 1967, Ackers, Harrison and Brewer 1967, Reddy and Pickford 1972, Halliwell and Saul 1980 and Saul and Delo 1981. However, most previous researchers have confined themselves to investigating the performance of overflows under steady state conditions or at most trapezoidal shaped storms. Thus to relate the performance of a hydraulic scale model to a prototype chamber it was considered pertinent to test chamber performance under unsteady flow conditions. This paper describes the development of the laboratory system and associated software necessary to generate an accurate and repeatable time discharge hydrograph of any profile, volume

or duration, based on the storm profiles outlined in the Flood Studies Report 1975, and to monitor the hydraulic performance of a double high side weir chamber in such a way that comparisons of chamber design may be made.

MODEL/PROTOTYPE SIMILARITY

To ensure that the model test results were representative of the prototype situation the geometry of the chamber was scaled by the linear scale ratio α where α is the ratio of prototype dimension to model dimension. As the flow in the model had a free surface all other parameters were scaled according to the Froude law ie the ratio of inertia to gravity forces. This gave the following scale ratios ; prototype discharge to model discharge $\alpha^{2.5}$, and prototype time to model time $\alpha^{0.5}$.

HARDWARE

The flow recirculation system and associated hardware are shown in Figure 1. A centrifugal pump raised the water from the sump to the constant head tank and the water was conveyed from this tank to the air operated butterfly valve through 76 mm diameter P.V.C. pipe. This pipe incorporated an orifice plate and a manual control valve. Downstream of the automatic valve the pipe increased in diameter to 112 mm perspex which formed the inlet pipe to the model and was 8m in length and set horizontal. The model itself was made from 6mm perspex and the chamber was constructed such that its geometry could be easily changed ie total length, weir length, weir crest height, width, baffle position etc. This enabled the influence of these parameters on the overflow performance to be readily assessed.

The high side weir chamber tested consisted of a stilling zone upstream of the weirs, the purpose of which is to allow the separation of gross sewage solids, followed by two weirs, one at either side of the chamber used to discharge the excess rainwater to the watercourse. This is followed by a storage volume which is employed in practice to retain the floating particulate and the pollutant load in the first foul flush. The dimensions of the tested chamber were as follows; overall length 5m, diameter of inlet pipe D = 112 mm, length of stilling zone LZ = 4.3D, weir length LW = 5.0D, length of storage volume LV = 35.3D, chamber breadth 1.4D and chamber weir height 0.9D. Scumboards were located 0.1D below weir crest and at a transverse distance 0.1D from the weirs. The chamber floor was benched at 1 in 4 and the flowrate to treatment was controlled by a penstock. The throttle flow and weir flow from the chamber were discharged to separate collection tanks each supported on an Avery weighbridge. This arrangement allowed steady flowrate to be accurately determined during calibration of the system.

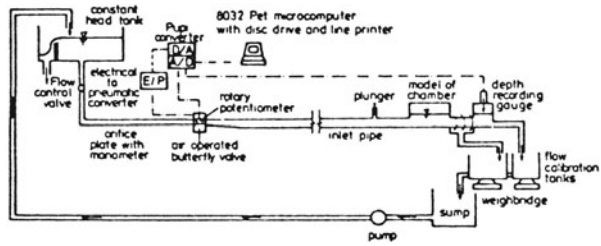
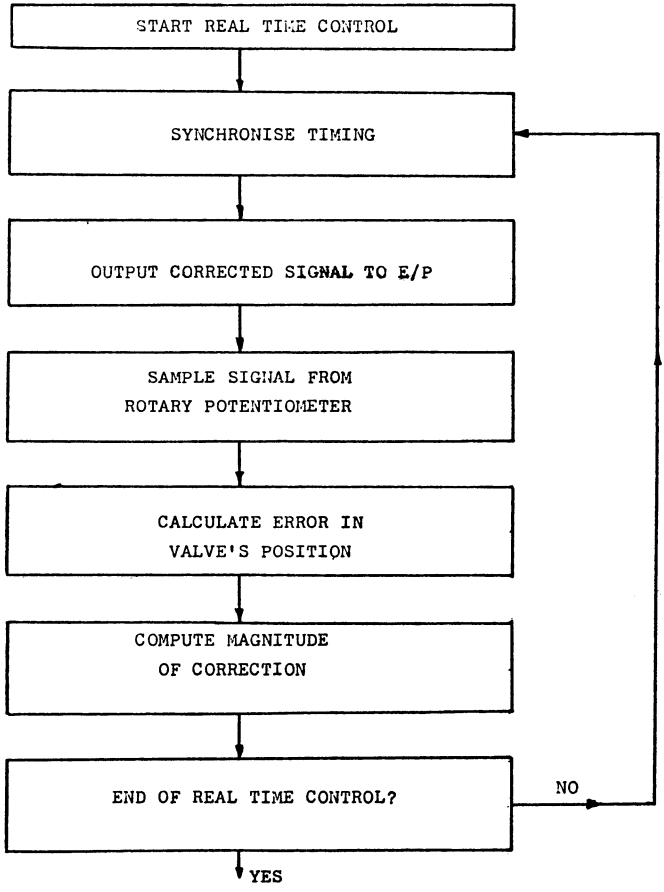


FIGURE 1 LAYOUT OF LABORATORY SYSTEM



FLOW CHART 1 REAL TIME CONTROL PROCEDURE

The microcomputer selected to control the system was a CBM 8032 which has the advantages of a 80 column integral memory mapped display, 32K of RAM, a user port for interfacing, an IEEE output bus and the availability of a large range of custom designed hardware and software. The dual disk drive used was the CBM 4040 which provided 170K of storage on each disk and a CBM 4022 dot matrix printer was used for hard copy output.

A Hewlett Packard A3 size digital plotter was used for graphics output and, as part of the software a general purpose plotting program GPLOT was developed to interface any HP-GL plotter to a series 32 CBM.

The interface used for the micro was a CIL 'Pupi" analogue/digital converter which was designed specically for the CBM. It was a 12 bit converter and had 4 analogue inputs ($\pm$ 10v),2 analogue outputs ($\pm$ 10v), 4 logic inputs and 4 relay closures. Only the analogue channels were used for control purposes. Its ROM based software provided a link to integer variables in BASIC programs which gave extremely simple operation of the interface. Due to the fact that all 14 channels were updated each time a conversion was called, the cycle time was relatively slow at 20 µs. However, this was of no significant disadvantage in the control of the valve.

The automatic valve was an air operated fish tail butterfly valve with a span of 90 degrees. The supply pressure was 18 psi while the input signal has a range 3-15 psi. To generate this input signal an electrical to pneumatic transducer (E/P) was utilised which gave a linearly varying output of 3-15 psi from an input signal of 1-5 mA supplied directly from an analogue output on the Pupi.

The feedback sensor was a contactless magneto rotary potentiometer attached to the shaft of the value and the body casing which provided an accurate measurement of the angular position of the valve.

The water depth in the chamber was monitored with a water level recorder which maintained a sharp pointed probe in contact with the water surface. The recorder housed a small motor, card potentiometer and circuitry which enabled the motor to drive the probe down each time it failed to contact the water surface and vice versa. A sensitivity dial controlled the rate of response of the probe. As the probe moved in response to the water surface the potentiometer recorded a voltage (via the Pupi) proportional to the water level from which the water depth was computed using a predetermined steady state calibration of depth versus voltage.

CONCEPT OF REAL TIME OPERATION

To generate a controlled unsteady pipe flow if was necessary to use an air regulated butterfly valve operated by a fully programmable signal. A microcomputer was used to generate a real time electrical signal implemented by a digital to analogue converter. Feedback from a sensor positioned on the valve formed a closed control loop. The principal of real time operation was based on the exact positioning (angular opening) of the valve to give, with respect to time, the desired time discharge hydrograph through the valve. The relationship between the angular opening and flowrate having previously been established under steady state conditions. The way in which real time operation was achieved is now described in some detail. Reference may be made to Flow Chart 1.

SOFTWARE

The five major programs CONTROL, CALIB, STORM, DATABASE and GPLOT and their related files are shown in Figure 2. Each program has been designed to perform a special function and the structure of each program is now described.

CONTROL

CONTROL is the central program and its function is to initialize the control data, to execute the real time operation of the valve, to calculate depths and flowrates, to display and print all results and to save the data to the relevant files.

The initialization part of CONTROL calculates the required output to the E/P and required input from the rotary potentiometer at one second time intervals, such that these values may be used in the real time control of the valve. The profile, duration and magnitude of the desired time-discharge hydrograph are entered at the keyboard together with the names of the appropriate files holding the calibration equations which relate the output (to E/P) and input (to rotary potentiometer) signals to flowrate. From the hydrograph profile entered, the data describing hydrograph shape is read from the STORM. LIB relative file. The ordinates of flowrate at one second time intervals are then calculated according to the value of duration and magnitude. These ordinates are then used in conjunction with the calibration equations to generate the output and input signals over the complete duration of the time discharge hydrograph. These signals were held in memory and the function of the real time procedure was, for a particular angular opening of the valve, to implement the output signal to give the desired valve opening, and to correct this position by minimizing the difference between the sampled input signal from the rotary

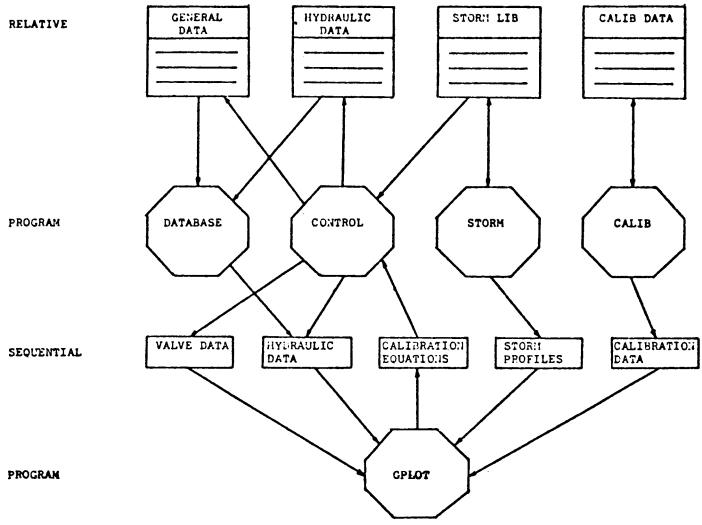


FIGURE 2 OVERALL STRUCTURE OF SOFTWARE

TEST NUMBER:03/10.4 NAME:95S/10/3.8 DATE:03/10/82

HYDROGRAPH PROFILE:95% SUMMER DURATION:10 M PEAK FLOW:3.8 L/S

CHAMBER GEOMETRY D=112 MM
LZ=4.3 D LW=5.0 D LV=35.3 D B=1.4 D C=0.9 D
RH=0.1 D RV=0.1 D BS= 1 ON 0 IC= 0 D

HYDRAULIC SETTING: QC=4.00 L/S QT=1.00 L/S

CALIBRATION CURVES:CAL1.1, CAL1.2, CAL2.3, CAL2.4, CAL2.5

CONTROL PARAMETERS:CF=3.5 TD=7 SECS NV=5

----------RESULTS----------

ERRORS(BITS):MR=1.01 MF=1.42 MT=1.21 SD=4.80 AR=-.12 AF=.13

TIME PERIOD	T	QE	QP	QC	QW	QT	QT/QC
TO PEAK	300	186	191	143	0	40	0.28
TO 1ST SPILL	310	218	225	168	0	49	0.29
SPILL	80	88	93	84	70	67	0.43
AFTER SPILL	430	31	29	93	0	163	1.75
TOTAL	820	337	347	345	70	279	0.81

TABLE 1 DISPLAY OF DATA FROM GENERAL DATA FILE

potentiometer and the desired input signal. The signal from the rotary potentiometer was based on a sample of 30 individual readings and obviously these readings were subject to a standard deviation error. To overcome this problem the correction to the subsequent output signal was calculated from a linear regression of the five previous values of the difference between the input computed from the calibration and the input signal sampled from the potentiometer.

Over the range of opening/closing operation of the valve the time constant of the valve was determined experimentally to be in the region 2 to 3 seconds. Hence to avoid instability in real time control of the angular position of the valve it was necessary to allow a period of time before making any correction to the position of the valve. In the experiments a time delay of 5 seconds was chosen during which time the error in the valve position was calculated and recorded but not corrected.

To determine the necessary correction signal to the subsequent (next) output signal it is necessary to use the calibration curves outlined in Figure 3, which relate output to input. For example, a one bit difference in the calibrated and sampled input gives a corresponding difference in flowrate and the gradient of the input calibration curve may be established for this difference. Similarly the gradient of the output calibration curve may be established over this range of flow-rate. The ratio of output to input gradient should then be used to correct the calibrated output signal. However, over the range of test conditions it was found adequate to simply multiply the input difference by 3.5 to determine the correction to the calibrated output signal.

The signal from the water level recorder was recorded at one second intervals and these signals were used to determine the flow depth in the chamber. The throttle flow and the over-flow weir flow were calculated using this depth from pre-determined steady flow calibration equations read off specific data files. The discharge time hydrograph at the valve was calculated at one second time intervals using the corrected position of the valve and a pre-determined steady state calibration curve.

The time discharge hydrograph input to the chamber was computed by storage routing the hydrograph at the valve through the 8m long inlet pipe assuming that the water level in the pipe and in the chamber was the same. The calculated hydrograph data was then displayed on the screen and/or printed.

The final part of CONTROL saves the data from the test run to the relevent files. General test data such as chamber

geometry, hydrograph description, calibration files used, results of error analysis and a synopsis of the hydraulic performance is written to the GENERAL DATA relative file whilst the detailed hydraulic data is written to the HYDRAULIC DATA relative file and also to individual sequential files for the purpose of plotting graphs using GPLOT.

CALIB

CALIB performs the precalibration of the system, maintains a relative file CALIB DATA of all calibration test data and creates sequential plotting files of any two parameters from a test for curve fitting with GPLOT.

The parameters recorded by CALIB are listed below.

1. Output signal to E/P (bits)
2. Input signal from Rotary Potentiometer (bits)
3. Angular position of valve - protractor (degrees)
4. Differential water head across orifice plate (mm)
5. Water depth in chamber (mm)
6. Input from water level recorder (bits)
7. Weir discharge (L/s)
8. Throttle discharge (L/s)

Figures 3 and 4 illustrate graphical examples of particular calibration tests created by CALIB.

STORM

STORM is used to create and store on the relative file STORM LIB the profile of each time-discharge hydrograph.

These hydrographs were based on data outlined in the Flood Studies Report 1975 and for the purpose of illustration the hydraulic performance of the chamber was tested using a 95% peakedness summer storm profile. The variation of discharge with time is outlined in Figure 5 prior to any modification caused by storage ie the discharge time hydrograph at the valve. STORM can also read, delete, display and print any individual profile and may be used to write sequential plotting files for any profile. The hydrograph is described by the coordinates and gradient of up to nine discrete points on the storm profile. A cubic curve equation is calculated between consecutive pairs of points and these equations together with the coordinates and gradients are stored on STORM LIB under the title given to the individual hydrograph.

DATABASE

DATABASE manages the two relative files containing all the data from every test run namely GENERAL DATA and HYDRAULIC DATA. It is used to search, read, display and print results of past

tests and can also produce sequential files of the data on the HYDRAULIC DATA file. This data may then be plotted with GPLOT. An example of the data saved to GENERAL DATA is shown in Table 1 in the format as displayed on the screen.

GPLOT

GPLOT is the general purpose plotting program which is used for all the graphical output of data recorded by the system. It is highly interactive menu driven program which runs on a series 32 CBM with disk drive and any Hewlett Packard graphics language (GL) digital plotter. All necessary parameters to describe a graph are initially keyed in under screen prompts after which they may be plotted, ammended, re plotted or saved to disk.

The parameters which describe a graph are the number of axes (2 or 3), numerical range of axes, the number of ticks on the axis, the annotation of the axes, type of data, source of data (keyboard or plotting file), character type and pen number of points (if necessary), line type and pen number (if necessary), data heading (if any), figure name and figure number which is used as the identification for saving the information to disk. The format of the graph can be altered by selecting any of the options; grid, key, box axes or border. There are three curve fitting options available, namely a straight line between consecutive points, a cubic curve between pairs of consecutive points and, a least squares polynominal curve fit based on discrete points.

When a graph is ready to be plotted the percentage A4 size must be specified followed by the section of the graph to be plotted ie the axes and annotation, the data groups (lines and/ or curves) or the whole graph. This allows curve fitting on new figures to be checked prior to the relatively slow operation of plotting the axes and annotation is performed.

RESULTS

To illustrate the performance of the system reference is made to the results of a specific test run and Table 2 records the listing of the error analysis and hydraulic data produced by CONTROL.

A measure of system accuracy may be obtained by comparing the storm volume of the desired hydrograph with that of the combined (summed) weir and throttle flow. The individual flowrate values listed in Table 1, when integrated over the complete period of the storm, give each component of storm volume. It can be seen that excellent agreement is obtained between the desired hydrograph volume of 337 litres and the combined weir and throttle flow volumes of 349 litres ie a 3% difference. Previous work Saul and Howarth 1982 has

TEST NO: 08/10.4
HYDROGRAPH: 95% SUMMER DURATION(MINS) 10 MAX 95% FLOW L/S 3.8
CORR. FACTOR TIME DELAY N
3.50 7 5
MEAN ERROR ABS.ERROR STANDARD DEV.
1.21 + .00 4.798
ABS ERROR (RISING) ABS ERROR (FALLING)
- .12 + .13

TIME	Q-PROG	Q-POT	Q-CHAM	D-CHAM	Q-WEIR	Q-THROT
20	.01	+ .03	+ .03	3.1	.00	.02
30	.01	+ .03	+ .02	3.1	.00	.02
40	.02	+ .02	+ .01	3.3	.00	.02
50	.03	+ .02	+ .02	3.3	.00	.02
60	.03	+ .04	+ .04	3.3	.00	.02
70	.04	+ .06	+ .05	3.4	.00	.02
80	.05	+ .05	+ .04	3.5	.00	.02
90	.06	+ .06	+ .05	3.8	.00	.02
100	.07	+ .08	+ .08	3.9	.00	.02
110	.09	+ .09	+ .09	3.9	.00	.02
120	.10	+ .12	+ .11	4.4	.00	.03
130	.12	+ .13	+ .12	4.7	.00	.03
140	.14	+ .16	+ .15	5.1	.00	.03
150	.17	+ .17	+ .16	5.6	.00	.03
160	.19	+ .19	+ .17	6.1	.00	.04
170	.22	+ .25	+ .22	7.0	.00	.04
180	.27	+ .28	+ .25	7.8	.00	.05
190	.32	+ .33	+ .30	8.5	.00	.05
200	.39	+ .39	+ .35	9.6	.00	.06
210	.48	+ .48	+ .40	10.8	.00	.07
220	.59	+ .57	+ .48	13.0	.00	.09
230	.72	+ .70	+ .58	15.2	.00	.10
240	.89	+ .87	+ .66	18.1	.00	.12
250	1.09	+1.07	+ .75	26.2	.00	.19
260	1.32	+1.33	+ .85	29.6	.00	.30
270	1.76	+1.99	+1.32	38.8	.00	.44
280	2.45	+2.57	+1.72	50.3	.00	.58
290	3.18	+3.13	+2.25	63.9	.00	.71
300	3.80	+3.90	+3.02	76.2	.00	.80
310	3.18	+3.39	+2.49	91.4	.00	.89
320	2.45	+2.62	+2.16	110.2	1.21	.97
330	1.76	+1.92	+1.92	114.1	2.09	.99
340	1.32	+1.39	+1.44	110.4	1.24	.97
350	1.09	+1.12	+1.15	105.1	.95	.97
360	.89	+ .90	+ .94	108.2	.75	.97
370	.72	+ .74	+ .78	107.1	.50	.96
380	.59	+ .61	+ .66	105.9	.24	.96
390	.48	+ .48	+ .59	104.2	.00	.95
400	.39	+ .39	+ .57	100.5	.00	.93
410	.32	+ .32	+ .55	96.1	.00	.91
420	.27	+ .26	+ .55	91.1	.00	.89
430	.22	+ .28	+ .50	85.6	.00	.86
440	.19	+ .19	+ .37	81.9	.00	.84
450	.17	+ .16	+ .35	79.6	.00	.83
460	.14	+ .13	+ .38	75.9	.00	.80
470	.12	+ .11	+ .40	71.8	.00	.77
480	.10	+ .09	+ .42	66.9	.00	.74
490	.09	+ .09	+ .42	62.0	.00	.70
500	.07	+ .07	+ .40	57.3	.00	.65
510	.06	+ .06	+ .35	52.4	.00	.60
520	.05	+ .03	+ .29	48.6	.00	.56
530	.04	+ .04	+ .20	44.8	.00	.52
540	.03	+ .02	+ .25	41.1	.00	.47
550	.03	+ .03	+ .23	37.9	.00	.43
560	.02	+ .01	+ .19	35.0	.00	.39
570	.01	+ .00	+ .16	32.3	.00	.35
580	.01	- .00	+ .15	29.7	.00	.30
590	.00	+ .00	+ .13	27.2	.00	.26
600	.00	+ .01	+ .12	25.1	.00	.23
610	.00	+ .00	+ .10	23.1	.00	.19
620	.00	+ .00	+ .09	21.4	.00	.16
630	.00	- .01	+ .07	19.7	.00	.13
640	.00	- .02	+ .04	18.1	.00	.12
650	.00	- .02	+ .08	16.8	.00	.11
660	.00	+ .00	+ .06	15.5	.00	.10
670	.00	- .00	+ .04	14.5	.00	.10
680	.00	+ .00	+ .04	13.4	.00	.09
690	.00	- .00	+ .03	12.8	.00	.08
700	.00	- .00	+ .03	12.0	.00	.08
710	.00	- .00	+ .01	11.2	.00	.07
720	.00	- .01	+ .01	10.7	.00	.07
730	.00	- .00	+ .01	10.1	.00	.07
740	.00	- .00	+ .01	9.6	.00	.06
750	.00	+ .00	+ .02	9.2	.00	.06
760	.00	+ .01	+ .02	8.7	.00	.06
770	.00	+ .00	+ .01	8.3	.00	.05
780	.00	- .00	+ .01	7.9	.00	.05
790	.00	+ .00	+ .01	7.6	.00	.05
800	.00	- .00	+ .00	7.3	.00	.05
810	.00	- .00	+ .00	7.0	.00	.04
820	.00	- .00	- .00	6.9	.00	.04
	337	347	345		70	279

TABLE 2 LISTING OF RESULTS AND ERROR ANALYSIS BY CONTROL

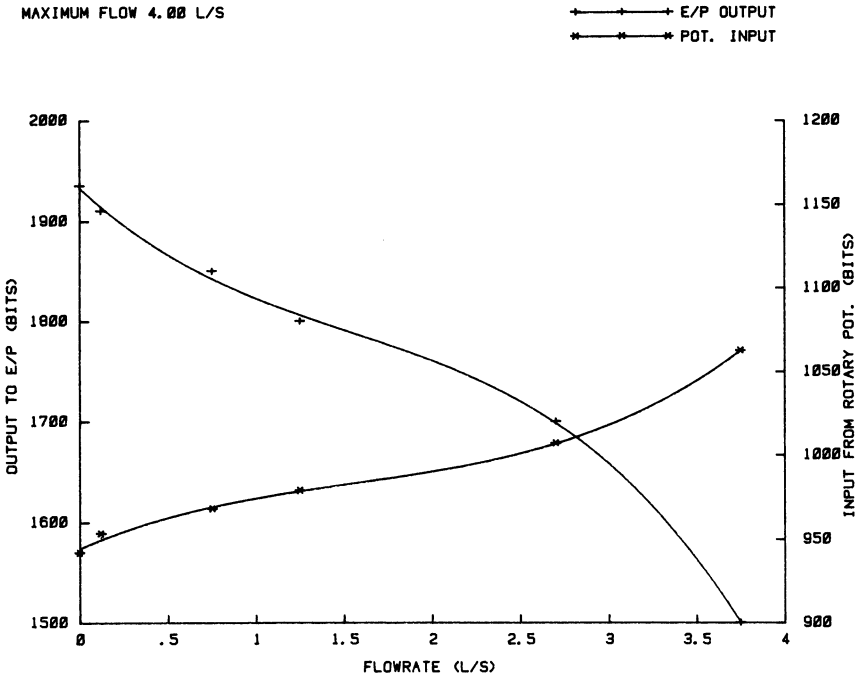


FIGURE 3 CALIBRATION OF VALVE'S CHARACTERISTICS

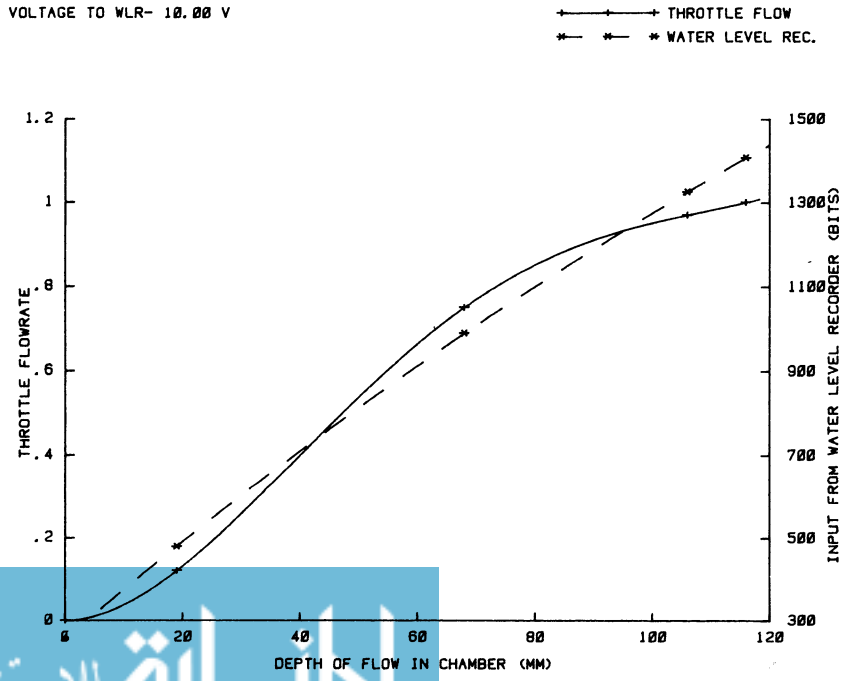


FIGURE 4 CALIBRATION OF WATER LEVEL RECORDER AND THROTTLE FLOW

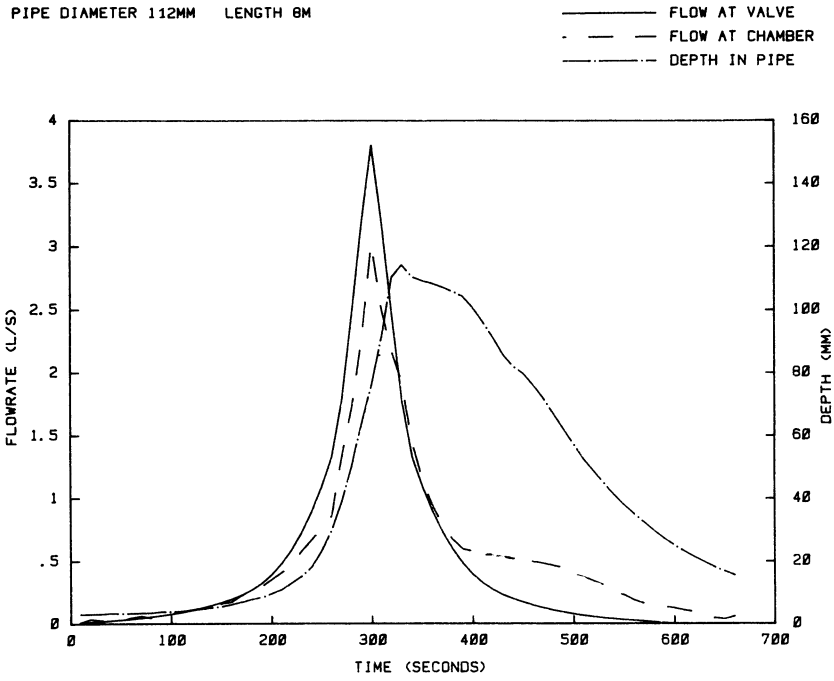


FIGURE 5 ATTENUATION OF HYDROGRAPH IN INLET PIPE

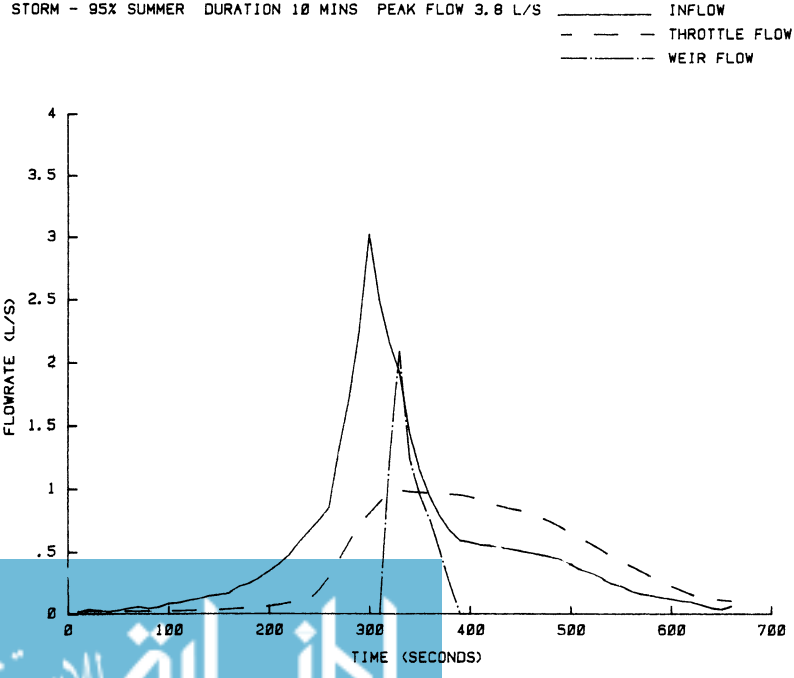


FIGURE 6 HYDRAULIC PERFORMANCE OF A DOUBLE HIGH SIDE WEIR OVEFLOW

illustrated that the system also gives excellent repeatability.

Figure 5 and 6 are included to demonstrate the graphical output by GPLOT. It is clear that parameters such as the influence of inlet pipe storage (Figure 5) and the hydraulic performance of a specific design of chamber (Figure 6) may be observed at a glance. As such the system may easily and quickly be used to test the hydraulic performance of different designs and geometric configurations of storm water overflow chamber. Conversely it is possible to optimize the dimensions of a particular type of chamber such that the desired hydraulic characteristics are established. Hence the above results and discussion demonstrate quite clearly the successful application of engineering software to a practical engineering problem.

ACKNOWLDGEMENTS

The work was carried out within the Simon Engineering Laboratories at the University of Manchester and the authors wish to thank the Science and Engineering Research Council for their financial support.

REFERENCES

1. Ackers, P., Harrison, A.J.M., and Brewer, A.J., "Laboratory studies of storm overflows with unsteady flow" Paper 4 I.C.E. Symposium on Storm Sewage Overflows 1967.

2. Frederick, M.R., and Markland, E., "The performance of stilling ponds in handling solids" Paper 5 I.C.E. Symposium on Storm Sewage Overflows 1967.

3. Halliwell, A.R., and Saul, A.J. "The use of hydraulic models to examine the performance of storm-sewage overflows". Proc. Instn. Civ. Engrs, Part 2, 69, June 1980.

4. Reddy, Y.R., and Pickford, J. "Stilling pond storm overflow studies", Jour. Sanitary Eng.Div. Proc. A.S.C.E., Vol 98, SA4, August 1972.

5. Sharpe, D.E. and Kirkbride, T.W., "Storm-water overflows; the operation and design of a stilling pond". Proc. Instn. Civ. Engrs, 13, 445-466, 1959.

6. Saul, A.J. and Delo, E.A., "Performance of a high-side weir storm sewage chamber incorporating storage. Proc. 2nd International Conf. Urban Storm Drainage, Illinois, June, 1981.

7. Saul, A.J. and Howarth, D.A., "Experimental study of discharge-time hydrographs in pipes". Sept. 1982. Proc.1st International Sem.Urban Drainage Systems,Southampton,

SIMULATION

A THEORIC-EXPERIMENTAL COMPUTER AIDED METHOD FOR THE PREDICTION OF OVERHEAD LINE DYNAMIC BEHAVIOUR AT HIGH SPEED.

M.No, R.Muguerza, L.Carrascosa,J.G.Giménez y J.García de Jalón.

Department of Applied Mechanics
University of Navarra.Urdaneta, 7 -SAN SEBASTIAN
SPAIN

ABSTRACT

One of the most relevant problems in high speed trains is the quality of contact between the overhead line and the pantograph. The loss of contact due to dynamic interaction causes damage in power units and wear in the line.

A great amount of work has been done during the last two decades by most important railway administrations to predict current collection systems behaviour.

This paper presents a mathematical model for computer simulation of this problem.

First of all, the overhead and the pantograph are analysed separately, for the validation of the developed mathematical models for each one. This checking includes linear and non linear behaviour, and wave propagation along the overhead line, comparing the theoretical and experimental results.

The overhead line is studied by means of matrix methods. The described model overcomes most of the principal deficiencies of the previous ones, taking into account, amongs other things, the following facts: flexural stiffness of the droppers beyond the pretraction load, and the possibility of a double contact wire.

A seven degree of freedom model has been developed for the pantograph. The experimental transfer functions have been fitted in a range from 0 to 30 Hz.

The representative equations of the overhead-pantograph system are integrated in time by an implicit method. Contact condition between the pantograph and wire(s) is taken into account by means

of Lagrange multipliers, whose physical meaning is that of the force of contact between them both.

In addition to previously mentioned facts, the present method takes into account the following ones: displacements of the pantograph base, contact prediction, and droppers desconexion (very important at high speeds).

All theoretical results are compared to experimental measurement.

INTRODUCTION

During recent years, many different studies on the overhead line-pantograph system have been carried out by most of the important railway administrations. The aim of these studies has been the development of a mathematical model which could allow a systematic, rational and economical optimization of the overhead-pantograph system. It is commonly accepted that the best way to do this is to analyze the dynamic behaviour of the overhead line and the pantograph separately, for the validation of the developed mathematical model for each one, and to join them into the coupled overhead-pantograph system, whose study is the objective of this kind of analysis.

This paper presents a mathematical model which allows the study of the different characteristic of the overhead and the pantograph, and the system behaviour at high circulation speeds, in a more general and deeper way than in previous ones. With this model it is possible to evaluate the influence that the different design parameters have on the mentioned dynamic behaviour.

The following paragraphs describe the mathematical model developed by the authors, emphasizing the most original and relevant aspects in relation to previous models.

ANALYSIS OF THE OVERHEAD LINE

The flexibility of the overhead line along the different spans may be evaluated by means of static measurements - theoric and/or experimental -. If the flexibility of the overhead line is very variable in a span, the contact force of the pantograph will cause different displacements in different points along the span, and this fact will induce elastic waves, oscilations and loss of contact.

The real behaviour of the catenary is non-linear. Droppers disposition is, in most cases, of such a kind that they cannot support compression loads. In fact, initially, droppers are tractioned by the pretraction of both catenary and contact wires, and thus they can support a compression load equal to the pretraction caused by this fact.

The Stiffness Method has been used for the modelization of the overhead line. In almost every previous model pin joined elements have been used solely. In this work beam elements have been used for two reasons: first of all, in the D.C. catenary used in Spain the wires are stiffer than the european ones; secondly, it seems difficult to evaluate accurately the force and the loss of contact between catenary and pantograph with linear elements (pin elements) of several meters of length. Pretractioned beam elements have been used for the catenary and contact wires, and pin joined elements for droppers. The number of spans considered in the analysis can be as large as required. For sixteen spans, which is the usual number in a section of the spanish overhead line, it results in a model of 2412 d.o.f. It has also been considered another particularity of the D.C. spanish catenary, which is the double contact wire.

Three different types of tests have been carried out for the validation of the mathematical model; two of them corresponding to the statical flexibility and the third referred to the elastic wave propagation. The first of them, perfectly linear, was made loading the overhead line with a vertical descending force. The second type of statical test is both more difficult and more interesting: the two contact wires were loaded simultaneously in ascending direction, in such a way that the displacement of the two wires were the same in the load application point. This last test has more interest because it is the way in which the pantograph loads the catenary, and is more difficult because the overhead line behaviour is non-linear.

It can be deduced from the experiments carried out, from the bibliography and from the overhead line characteristics, that the damping is quite small in the frequency range of interest for this study. It has been estimated that this value lays between 0,1% and 2% in the range from 0 to 30 Hz. In this conditions the damping can be considered as a viscous proportional damping, and the damping matrix $|C|$ takes the form

$$|C| = \alpha_o |M| + \alpha_1 |K| \qquad (1)$$

where $|M|$ and $|K|$ are the mass and stifness matrices, and α_o and α_1 are the proportionality coefficients which must be evaluated to obtain a modal damping between the previously mentioned values.

The mathematic model has been used for the evaluation of the natural frequencies and shapes of vibration, and for the study of the propagation of elastic waves. The comprehension of the number and distribution of the natural frequencies of the catenary has been very relevant, in this study, in order to decide the best way to overtake the analysis of the system overhead-pantograph.

In fact, the great number of natural frequencies of the catenary (1.317 below 30 Hz) and their very dense distribution (in groups of 33 in a 1/10 Hz interval),together with another reasons mentioned later, made the authors discard the use of a modal basis for numerical integration of the differential equations of the system. For these reasons the experimental modal analysis of the overhead-line was also discarded.

For the dynamic analysis of the pantograph a method has been used different from the one used in the dynamic analysis of the catenary, because it has been proved difficult to develop an accurate mathematical model for the pantograph. This difficulty arises from the non-linearities: Coulomb friction, clearances, and the existence of large rigid solid displacements together with small elastic deformations. For these reasons the pantographs mathematical model has been developed using experimental modal analysis.

This method consists in, assuming an initial configuration of the model to represent the pantograph, to fit the transfer functions of that configuration to those obtained by an experimental modal analysis. This fitting,made by a least squares method,permits the determination of the mathematical model's parameters.

Taking into account that the peak values which appear in the transfer functions correspond to the natural frequencies,it will be taken as the n.d.o.f. for the mathematical model the number of the more significant peaks which appear in the experimental transfer functions in the frequency range of interest.

The developed method permits the use of a number of d.o.f. larger than previous models (usually two), which correspond to the more significant natural frequencies of vibration.

Assuming proportional damping in the pantographs mathematical model, the dynamic equilibrium equations in modal coordinates can be established

$$|m|\ \{\ddot{x}\} + |c|\ .\ \{\dot{x}\} + |k|\ \{x\} = |R|^T\ \{f\} \qquad (2)$$

where

- $|m|$, $|c|$ and $|k|$ are diagonal mass, damping and stiffness modal matrices
- $\{f\}$ is the vector of applied loads
- $|R|$ is the modal matrix (rectangular matrix whose columns are the natural shapes of vibration used as a basis), according to the idea of this method, the coefficients of $|m|$, $|c|$ and $|k|$ matrices are the values to be fitted.

Obtaining the Fourier transform in equation (2), and solving into the term belonging to the vertical movement of the head of

the pantograph (which is the one that appears in the loss of contact phenomenon), it becomes

$$X_1 = \Sigma \frac{\omega^2 \{r^i\}^T \left|\bar{M}\right| \{X_b\}}{m_i(-\omega^2+2j\xi_i\omega_i\omega+\omega_i^2)} + \Sigma \frac{F_c}{m_i(-\omega^2+2j\xi_i\omega_i\omega+\omega_i^2)} \qquad (3)$$

or in a more condensed form

$$X_1 = \{H_{cb}(\omega)\}^T \{X_b\} + H_{cc}(\omega).F_c \qquad (4)$$

where: (being $j^2=-1$)

- $\{r^i\}$ is the i-th natural shape of vibration
- $\omega_i^2 = \frac{k_i}{m_i}$ and $c_i = 2\xi_i\omega_i m_i$ are the coefficients to be fitted
- $\{X_b\}$ is the Fourier transform of the vector which represents the 6 d.o.f. of the pantograph base.
- $\left|\bar{M}\right|$ is a mass matrix which gets the inertia forces due to the pantograph base movement.
- F_c is the Fourier transform of the force applied to the head of the pantograph.

Equations (3) and (4) reflect the influence that the two entries of the system, the contact force on the head and the displacement of the base, have on the movement of the pantograph head.

In equation (4) the term $H_{cc}(\omega)$ is easily obtained by an experimental analysis in which the pantograph's base is fixed to the ground ($\{X_b\} = \{\emptyset\}$). However, the direct experimental evaluation of the transfer function $H_{cb}(\omega)$ would require a complicated experimental equipment which could permit the excitation of the base with a known movement. To overlay this difficulty, the following indirect method has been used.

The pantograph's base has been supported on a set of soft springs, and the function $\bar{H}_{cc}(\omega)$ is evaluated ($\bar{H}_{cc}$ is different from H_{cc} in the sense that the former is evaluated with the pantograph base) movable.

$$X_1 = \bar{H}_{cc}(\omega) \cdot F_c \qquad (5)$$

it is also possible to evaluate the transfer function between pantograph's top and base

$$\{X_b\} = \{\bar{H}_{bc}(\omega)\} \cdot F_c \qquad (6)$$

sustituting (6) in (4), equating to (5), and reordering, we arrive at

$$\{H_{bc}(\omega)\}^T \{H_{cb}(\omega)\} = \bar{H}_{cc}(\omega) - H_{cc}(\omega) \qquad (7)$$

but these equations are not enough to determinate $\{H_{cb}(\omega)\}$. This new difficulty is solved by shaking in other points, subindicated with s, and writing a new set of equations

$$\{\bar{H}_{bs}(\omega)\}^T \{H_{cb}(\omega)\} = \bar{H}_{cs}(\omega) - H_{cs}(\omega) \qquad (8)$$

In order to obtain a determinated system from equations (7) and (8), it is necessary to shake the pantograph's base in a number of points equal to the n.d.o.f (number of components of the $\{H_{cb}(\omega)\}$ vector); however it is better to do it in a greater number of the s points than strictly necessary and to solve an overdeterminated system via the least squares method.

Once the transfer functions $\{H_{cb}(\omega)\}$ and $H_{cc}(\omega)$ of expression (4) have been evaluated experimentally by the described method, they are fitted into expression (3) determining the parameters of this last equations, and then the theoretical mathematical model is definite.

DYNAMIC ANALYSIS OF THE OVERHEAD LINE-PANTOGRAPH SYSTEM

The problem of the dynamic behaviour of the catenary-pantograph system is a problem of dynamic interaction between two systems whose relative position changes in the time domain at a known rate. The bibliography on this topic is not very extensive, and a good revision can be found in the bibliographic references.

The methods based on a change to modal coordinates have three major difficulties which make the methods not viable in this particular problem. The first one is the grant number of natural frequencies that the catenary has in the frequency range of interest. Likewise, the presence of a movable element - the pantograph - causes the system of differential equation to be coupled, also in the modal basis. Finally, the modal basis does not allow to take into account the non-linear behaviour of the overhead line.

An important aspect in the numerical integration is the non-linearity caused by the droppers behaviour previously mentioned. This fact has been taken into account by means of the Residual Forces Method. This method has a major advantage that makes it very useful in this case. It is not necessary to recalculate

the stiffness matrix at each stage, an iterative correction, which only implies some back substitutions,is enough.

An implicit method has been selected for the numerical integration of system of differential equations. There are a lot of numerical integration implicit methods, either for first order systems or for second order ones. The method selected by the authors is based on Jensen's idea. This idea consists on applying a method suitable for first order differential equations to a second order system, and doing it without increasing the number of equations. Felippa and Park have proved the superiority of this formulation, referring to errors propagation, over the mentioned second order methods.

Lagrange equations, with restrictions and Lagrange multipliers have been used for the establishment of the motion system of differential equations, because they allow a great simplicity for the introduction of restrictions. In this case, restrictions must reflect the condition of equal displacement for the head of the pantograph and the two contact wires. Thus the resulting system of differential equations takes the form,

$$\begin{bmatrix} |M_c| & |0| & |0| \\ |0| & |m_p| & |0| \\ |0| & |0| & \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix} \end{bmatrix} \begin{Bmatrix} \{\ddot{x}_c\} \\ \{\ddot{\tilde{x}}_p\} \\ \ddot{\lambda}_1 \\ \ddot{\lambda}_2 \end{Bmatrix} +$$

$$\begin{bmatrix} |C_c| & |0| & |0| \\ |0| & |c_p| & |0| \\ |0| & |0| & \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix} \end{bmatrix} \begin{Bmatrix} \{\dot{x}_c\} \\ \{\dot{x}_p\} \\ \dot{\lambda}_1 \\ \dot{\lambda}_1 \end{Bmatrix} +$$

$$\begin{bmatrix} |k_c| & |0| & \{N_1(t)\}\{N_2(t)\} \\ |0| & |k_p| & |-1| \\ \{N_1(t)\}^T & |-1|^T & \begin{bmatrix} -k_1 & 0 \\ 0 & -k_2 \end{bmatrix} \end{bmatrix} \begin{Bmatrix} \{x_c\} \\ \{x_p\} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \tag{9}$$

$$= \begin{Bmatrix} \{0\} \\ \begin{Bmatrix} -F \\ -F \end{Bmatrix} + \{\tilde{p}(t)\} \\ f(vt) \\ f(vt) \end{Bmatrix}$$

where:

- $|M_c|$, $|C_c|$, $|K_c|$, $|m_p|$, $|c_p|$ and $|k_p|$ are the mass, damping and stiffness matrices of the overhead and pantograph, respectively (diagonal matrices for the pantograph).
- $\{X_c\}$ and $\{X_p\}$ are the vectores of generalized coordinates of the overhead and modal coordinates of the pantograph, respectively.
- $\{N_1(t)\}$ and $\{N_2(t)\}$ are the vectors of shape functions of the overhead line evaluated at the point of contact.
- k_1 and k_2 are proportionality coefficient used to calculate the double clamped displacements in the loaded elements.
- F is the static force between overhead-line and pantograph (including aerodinamic force).
- $\{\tilde{p}(t)\}$ represents the effect of the base movement of the pantograph.
- f(vt) introduces the initial deflection if the overhead line in the equations of restriction.
- λ_1 and λ_2 are the Lagrange multipliers, whose physical meaning is that of the force of contact between pantograph and each of the contact wires, with changed sign.

These equations have the following characteristics:

1) All the matrices appearing in (9) are symmetric (but non positive-definite).
2) Only one of these three matrices has terms which depend on the time, and these terms are concentrated in the two last rows and columns.
3) The phenomenon of the loss of contact between the pantograph and one (or two) of the contact wires is easily detected, because it happens when one (or two) of the Lagrange multipliers becomes positive. At this moment, equations (9) are still valid only by eliminating the corresponding row(s) and column(s) in the restriction equation(s), while the contact is again detected.

RESULTS AND CONCLUDING REMARKS

Figure 1 shows one of the graphics of flexibility deduced from the linear tests (vertical descending force). A very good agreement has been obtained between both the theoretical and experimental results (errors less than 8%). The presence of a peak of rigidity is noted on the points where catenary wire joins the Y dropper.

The main and final aim of this work has been reached by doing a

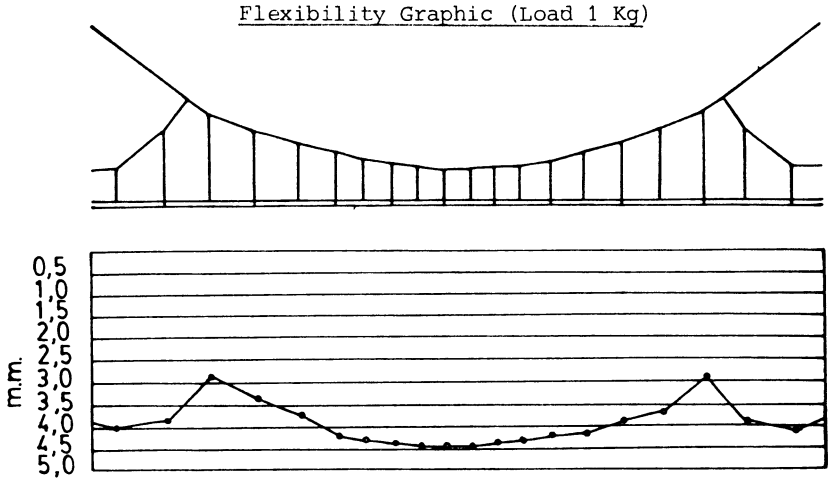


Fig.1.- Flexibility graphic, lineal test

set of computer program runs, for different circulation speeds of the train, and taking into account facts such as the catenary initial deflection, droppers' non-linearity, and the rail and locomotive conditions (by means of the movement of the pantograph's base).

Figure 2 shows the graphic output such as it is done by the computer. Table 1 summarizes the numerical results of some runs. The two entries are the train speed and the number of losses of contact between the pantograph and the two contact wires in the run along the first five spans of a overhead line section. The following values are represented in this figure:

- F_1 = contact force in wire 1 (in N.)
- F_2 = contact force in wire 2 (in N.)
- F_t = total contact force (in N.)
- D_1 = vertical displacement of wire 1 (in cm)
- D_2 = vertical displacement of wire 2 (in cm)
- D_p = vertical displacement of pantograph's head (in cm)
- F_1 = initial deflection of the overhead (experimental values) (in cm)
- D_b = vertical displacement of the pantograph's base (experimental values) (in cm)

The step size used has been 22.5 cm, changing the corresponding time step according to the train speed.

The value of the pretension load, and the ones of the static and aerodinamic contact forces have been estimated following

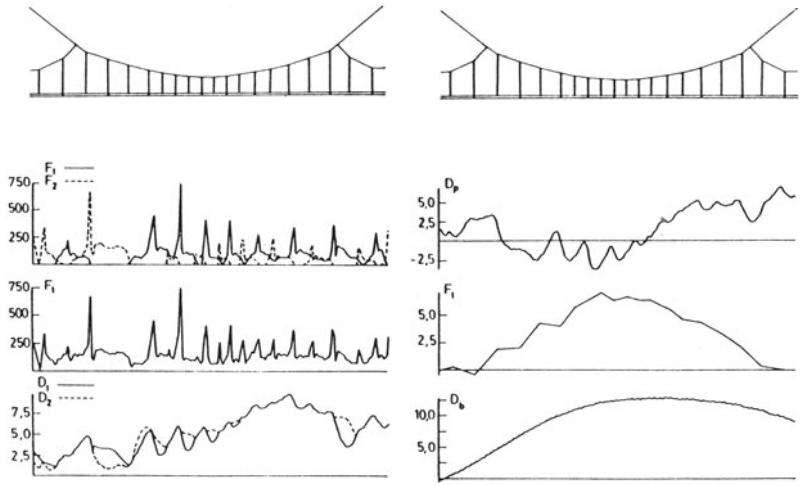


Fig.2.- Computers' graphic output

spanish railway administration (RENFE) specifications.

Table 1.- Nº of losses of contact in five spans

Speed Km/h	120	140	160	180	190	200
Nº of losses of contact	0	0	1 0*	3	3	6 4*

(*) With the pretractions increased

It can be concluded that the present mathematical model considerably improves the reliability of the result in relation with previous models; and it becomes a more efficient working tool, because it allows the different design parameters in the whole system behaviour to be taken into account. In the near future the present authors hope to solve the problem of the catenary-pantograph system optimization, and also hope to develop a simple and reliable method to count the number and duration of the losses of contact.

REFERENCES

Bathe, K.J. and Wilson, E.L. (1976) Numerical Methods in Finite Element Analysis. Prentice Hall.

Beaufait, F.W. Rowan, W.H.Jr. Hoadley,P.G. and Hackett, R.M. (1970) Computer Methods of Structural Analysis. Prentice Hall.

Boissonade, P. and Dupont, R. (1977) Current Collection with Two-Stage Pantographs on the New Paris-Lyon. Railway Gazette International.

Felippa, C.A. and Park, K.C. (1978) Computational Aspects of Time Integration Procedures in Structural Dynamics-Part 1:Implementation. ASME Journal on Applied Mechanics, 8:595.

Fischer. (1977) Le Caténaire et le Pantographe à Très Grande Vitesse. ZEV-Galsers Annalen, 5.

García de Jalón, J. Giménez, J.G. Nó M. y otros. (1981) Teoría y Práctica del Análisis Modal. Escuela Superior de Ingenieros Industriales de San Sebastián.

Gear, C. (1971) Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall.

Gladwell, I. and Sayers, D.K. (1980) Computational Techniques for Ordinary Differential Equations. Academic Press.

Giménez, J.G. (1979) Recientes Avances en el Análisis Modal Experimental de Sistemas Mecánicos. Jordanas Técnicas sobre Vibraciones.

Giménez, J.G. y Carrascosa, L.I. (1982) Global Fitting: An Efficient Method of Experimental Modal Analysis of Mechanical Systems. First International Modal Analysis Conference.

Gostling, R.J. and Hobbs, A.B.W. (1981) The interaction of Pantographs and Overhead Equipment: Practical Applications of a New Theoretical Method. Mechanical Engineering.

Henrici, P. (1962) Discrete Variable Methods in Ordinary Differential Equations. Wiley.

Hilber, H.M. Hughes, T.J.R. and Taylor, R.L. (1977) Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics. Earthquake Engineering and Structural Dynamics, 5:283.

Hilber, H.M. and Hughes, T.J.R. (1978) Collocation, Dissipation and Overshoot for Time Integration Shemes in Structural Dynamics. Eartquake Engineering and Structural Dynamics. 6:99.

Hobbs, A.E.W. (1977) Accurate Prediction of Overhead Line Behaviour. Railway Gazette International.

Huang, T. (1976) Vibration of Bridges. Shock and Vibration Digest, 8: 61.

Jensen, P.S. (1974) Transient Analysis of Structures by Stiffly Stable Methods. International Journal of Computers and Structures, 4: 615.

Lambert, J.D. (1973) Computational Methods in Ordinary Differential Equations. Wiley.

Luppi, J. (1976) Captage à Grande Vitesse. Cours de Formation des Atachés et Dessinateurs. VZC.

Luppi, J. (1981) Captage du Couran á Grande Vitesse. Exposé au Comité Mixte de la Direction V.

Nó,M. Muguerza,R. Giménez, J.G. y García de Jalón, J. (1982). Estudio Dinámico a Velocidades Elevadas de la Catenaria Tipo Renfe Modernizada y Compensada. Escuela Superior de Ingenieros Industriales.

Nó, M. y García de Jalón, J. (1981) Vibraciones en Vigas Continuas Producidas por su Interacción con Sistemas Móviles. Aplicación al Sistema Ascensor-Guías. IV Congreso Nacional sobre la Teoría de Máquinas y Mscanismos.

Park, K.C. and Felippa, C.A. (1978) Computational Aspects of Time Integration Procedures in Structural Dynamics- Part 2: Error Propagation. ASME Journal on Applied Mechanics, 45:603.

"MICRO-COMPUTERS: A NEW SIMULATION PROGRAM FOR TRUCK TRAVEL TIME IN OPEN PIT MINES"

BY

T. Abdallah[1] and C. Chambon[2]

1) Dr.Ing.,Department of Mining,Petroleum and Metallurgical Engineering,Faculty of Eng.,Cairo University,Giza (Egypt)
2) Professor,Ecole Nationale Supérieure de la Métallurgie et de l'Industrie des Mines, (E.N.S.M.I.M),parc de saurupt, 54042 Nancy (France).

INTRODUCTION

Previous computer programs simulating materials handling in open pit mines use a well-established model for the description of vehicle motion [2,3,4]. This deterministic model, assuming a constant acceleration, relies on a characterestic performance curve which relates vehicle speed with the propelling force (Rim pull) available at the drive wheels. The characteristic curves are supplied by truck manufacturers. The model assumes that a truck driver will always use the maximum power available at a given speed following the characteristic curve. However, this last assumption is not necessarily realistic. Moreover, not all manufacturers supply characteristic curves for their vehicles. On the other hand, the model which is constructed for big computers needs a very long time of execution when it is adapted in micro-computers. There is thus a need for a new algorithm for the calculation of truck travel-time, which is not dependent on characteristic curves, and which takes into consideration the real power available to the truck. This new algorithm is essentially to be applied on a micro-computer (Apple II 48 K). A program, called CALCUL, is described in this paper. This program is compiled by the package TASC to reduce the time of execution.

THE ALGORITHM

The profile of haul road between loading and dumping points is taken to consist of a number of elements. Each element is defined by a length, d, a grade (in percent),ρ, and a curve speed, V_c, which is the maximum allowable speed at the end of the element, imposed for safety reasons on consideration of change of

grades or sharp corners. For first element, the initial speed is zero (truck at loading point). Also for the last one (truck at dumping point) the final speed is zero.

A fundamental hypothesis of the algorithm is that : trucks are traveling at constant power. This is realistically acceptable and is consistant with the characteristic curves supplied by the manufacturers.

$$W = FV \qquad \text{as} \qquad W = \zeta W_o \tag{1}$$

Where W_o is the nominal power, ζ is a power coefficient (about 80%), F is the rim pull, V is the truck speed and W is the real power.
Figure 1 shows that the constant power curve does not differ from the characteristic curve except when the speed is less than 5 km/h (region of small speed). Equation (1) gives an infinite rium pull for a zero speed. However, this can not invalidate the hypothesis since, in practice, a truck is traveling in this region only when it leaves or arrives at either loading or dumping point. The time required to reach a speed of 5 km/h is so small that its influence on the total time of traveling may be considered as negligible.

Newton's Second Law of Motion gives :

$$F - (\rho+r)\, P = \frac{P}{g}\, \gamma \tag{2}$$

where P is the truck weight (laden for haul and unladen for return), g is the gravitational acceleration, γ is the truck acceleration and r is the rolling resistance of the haul road in percent.

We define the equivalent grade $\alpha = \rho + r$

then $$\gamma = \frac{gW}{P}\, \frac{1}{V} - g\, \alpha \tag{3}$$

This equation will be examined for three cases.

1- <u>When $\alpha > 0$ (uphill)</u>:

Equation (3) can written as

$$V\, V' = \frac{gW}{P} - g\, \alpha\, V$$

Which may be integrated and gives with the boundary conditions ($t=0$, $x=0$, $V = V_c$):

$$\frac{gW}{P}\, t - g\, \alpha\, X = \frac{1}{2}\, (V^2 - V_c^2) \tag{4}$$

Also, equation (3) may be written as :

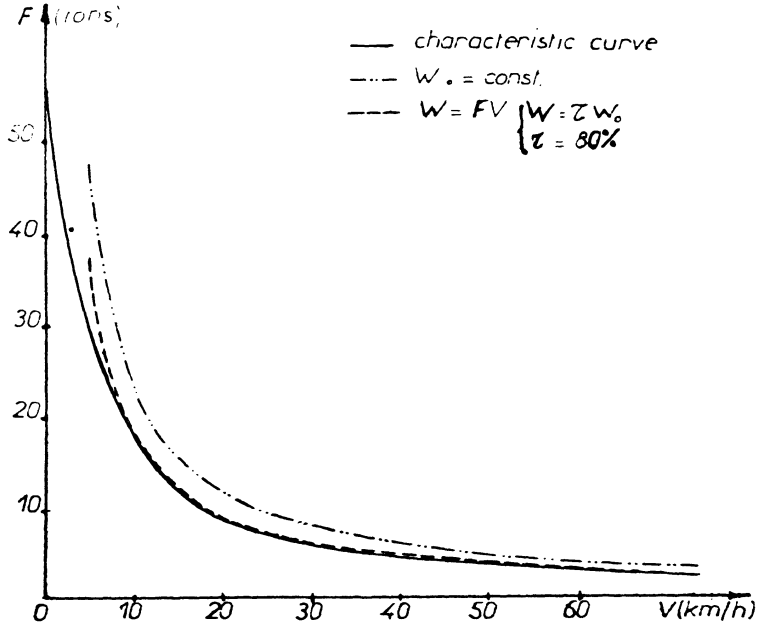


Figure 1 : Characteristic and constant power curves for a caterpillar 777.

$$\frac{V\ dV}{\frac{W}{\alpha P} - V} = \alpha\ g\ dt$$

The integration of this equation with the above boundary conditions gives :

$$\alpha\ g\ t = -(V-V_c) - \frac{W}{\alpha P} \ln \left[\frac{\frac{W}{\alpha P} - V}{\frac{W}{\alpha P} - V_c} \right] \qquad (5)$$

The term $W/\alpha P$ has the dimensions of speed and will be called, speed limit, V_ℓ

Hence $$V_\ell = \frac{W}{\alpha P} \qquad (6)$$

Substituting t in equation (4) using equation (5), then :

$$\left.\begin{aligned} X &= \frac{-1}{2\alpha g}(V^2-V_c^2) - \frac{V_\ell}{\alpha g}(V-V_c) - \frac{V_\ell^2}{\alpha g} \ln\left[\frac{V_\ell - V}{V_\ell - V_c}\right] \\ t &= \frac{-1}{\alpha g}(V-V_c) - \frac{V_\ell}{\alpha g} \ln\left[\frac{V_\ell - V}{V_\ell - V_c}\right] \end{aligned}\right\} \qquad (7)$$

These equations depend only upon three parameters α g, V_ℓ and V_c which are known for a given element of road. To calculate the distance and the time required to reach the speed limit, V_ℓ we assume an uniform acceleration of α g (see Appendix 1). Then, the time required $t(V_\ell)$ is given by :

$$t(V_\ell) = (V_\ell - V_c)/\alpha\ g \qquad (8)$$

Let $V = V_\ell$ and substitute t in equation (4) using equation (8). The distance traveled $X(V_\ell)$ is then given by :

$$X(V_\ell) = \frac{t(V_\ell)}{2} \cdot (V_\ell - V_c) \qquad (9)$$

2- When $\alpha = 0$ (flat slope):

With $\alpha = 0$ in equation (4), we get

$$V^2 - V_c^2 = \frac{2gW}{P} \cdot t \qquad (10)$$

We obtain by integration :

$$\frac{3gW}{P} X = \left(\frac{2gW}{P} t + V_c^2\right)^{3/2} - V_c^3 \qquad (11)$$

substituting t in equation (11) by equation (10), then :

$$X = \frac{P}{3gW} (V^3 - V_c^3) \qquad (12)$$

Equations (10) and (12) depend only upon two parameters gW/P and V_c which are known for a given element of road.

3- When $\alpha < 0$ (downhill):

The same procedures as for a zero grade are employed.

DESCRIPTION OF THE PROGRAM

The simulation is thus oriented in two directions according to the equivalent grade. Firstly for an uphill grade and secondly for either flat or down hill grade. An option on the maximum speed is adopted[1].

If the speed is not limited by the user, a value of 60 Km/h is chosen by the micro-computer. The curve speed, V_c, defined by the user also defines the final speed V_f at the end of the given element and will define subsequently the initial speed, V_o, for the next element. The procedures of the simulation are illustrated by the following examples :

a- If $\alpha > 0$:

1- The speed limit V_ℓ is computed using equation (6).

2- Let for example V_m be the maximum speed and $V_\ell \leqslant V_m$.

3- $t(V_\ell)$ and $X(V_\ell)$ are calculated for $V_c = V_o$ using equations (8) and (9). As V_o may be greater than V_ℓ the absolute difference is considered.

4- To simplify the example let $V_o = V_f$.

5- If $X(V_\ell) \leqslant d/2$, the travel time t is then :

$$t = 2 \left[t(V_\ell) + \frac{d/2 - X(V_\ell)}{V_\ell} \right] \qquad (13)$$

6- If $X(V_\ell) > d/2$, the speed limit, V_ℓ, will not be reached.

Then the truck speed, V, is calculated by an iterative method (Newton's Method). This iterative procedure stops when $|X-d| \leqslant 0.5$ (50 cm of precision). Hence, travel time is known.

Other cases concerning V_o,V_f and V_m with regard to V_ℓ are considered in the program as shown in figure 2a-2b.

b- If $\alpha \leqslant 0$

1. a value of maximum speed, V_m, is defined.
2. $t(V_m)$ and $X(V_m)$ which are respectively the time and the

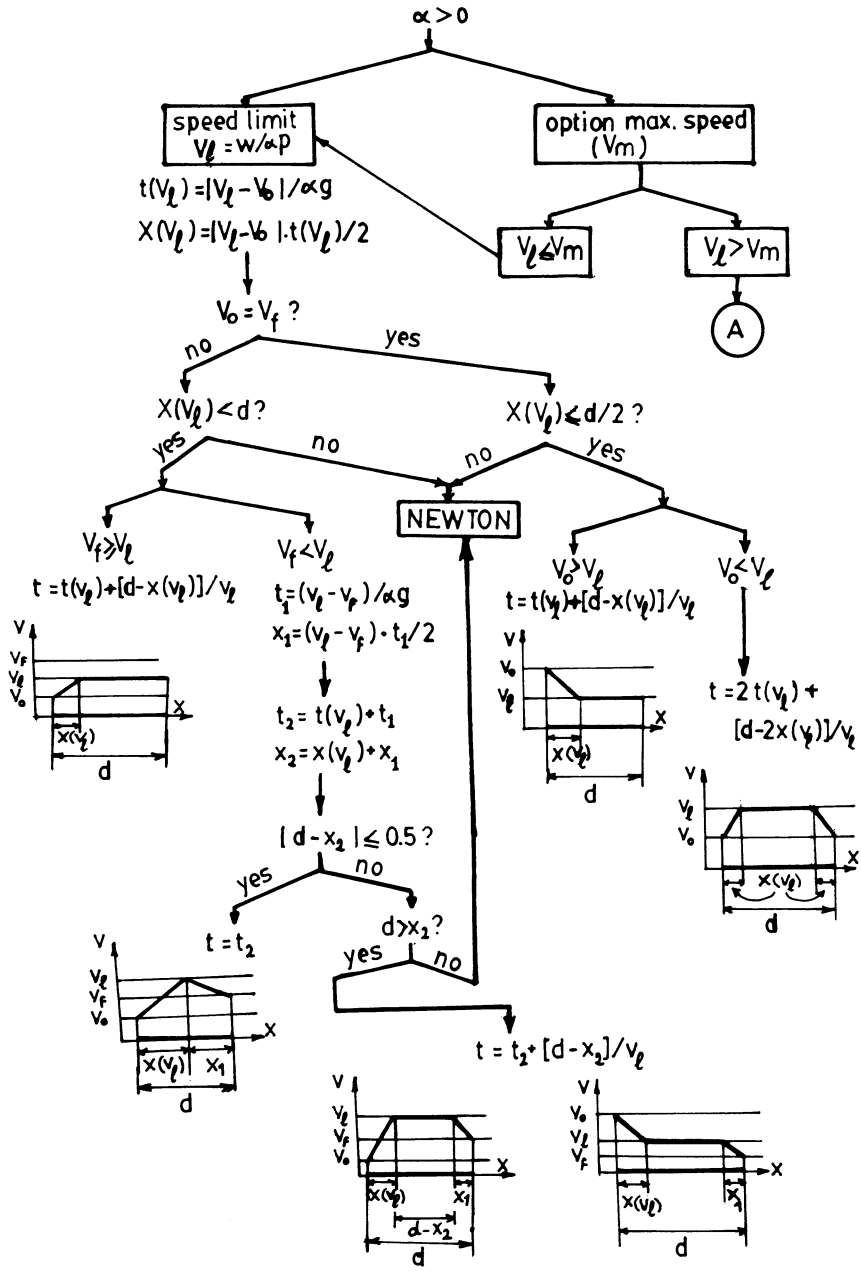


Figure 2a: Schema of travel time calculation for uphill grade.

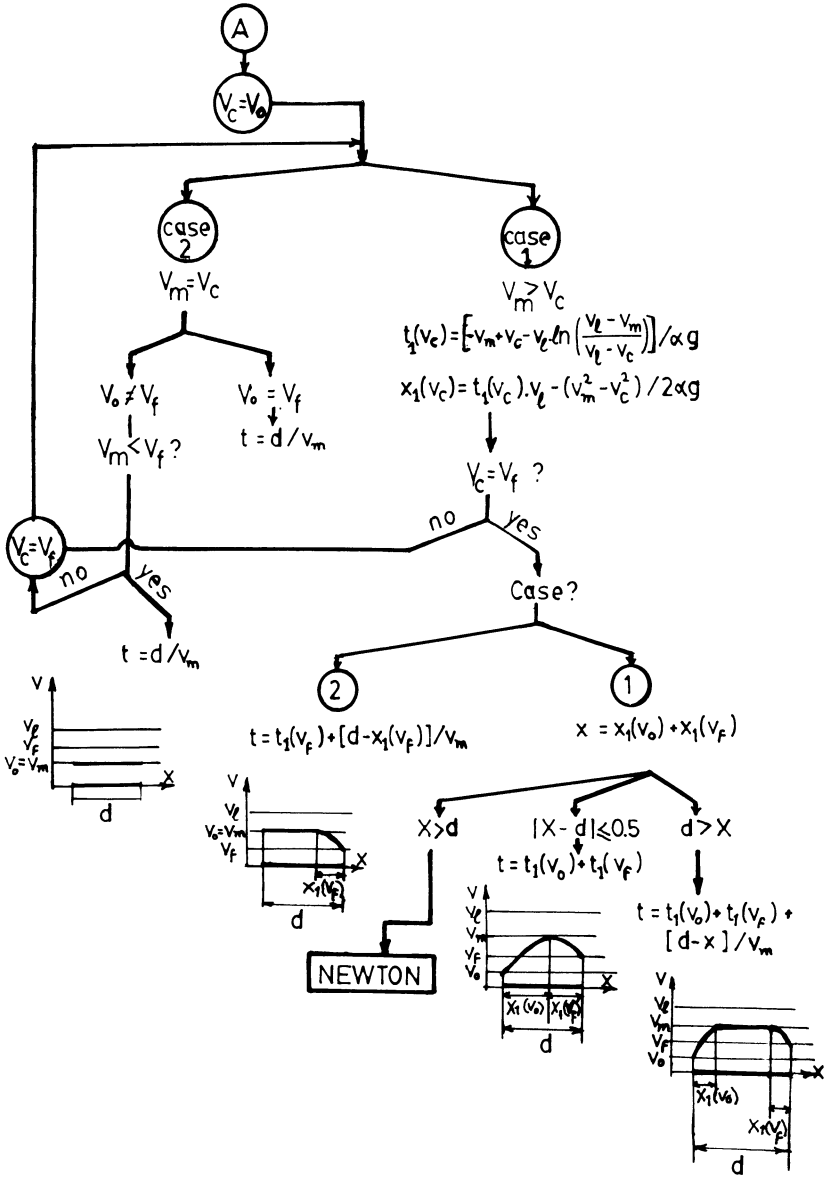


Figure 2 b

distance traveled to reach V_m are computed using equations (10) and (12) with $V = V_m$ and $V_c = V_o$.

3. to simplify the example let $V_o = V_f$.

4. if $X(V_m) \leq d/2$, the travel time, t ,is then

$$t = 2 \left[t(v_m) + \frac{d/2 - X(V_m)}{V_m} \right] \qquad (14)$$

5. if not an iterative procedure (Newton) gives the speed,V, of trucks reached on this element of road. The travel time t is then :

$$t = \frac{P}{gw} \cdot V^2 - \frac{P}{2gw} (V_o^2 + V_f^2) \qquad (15)$$

The schema of time calculation for this case is illustrated in figure 3.

At each option of speed (V_ℓ or V_m), the corresponding average speed is calculated to illustrate the influence of the option adopted on the speed of trucks. As the average speed is well known for an open pit mine, it will help the miner to choose the proper option.

CONCLUSIONS

The hypothesis of constant power better reflects the reality because it considers the real power used by the drivers which is proportional to the nominal power. The curves of constant power can replace the characteristic ones. This algorithm is faster when applied on micro-computers. Moreover, the time calculated by the program, CALCUL, has been compared with time observed at Decazeville surface coal mine in France. A good correlation between the two times has been found.

REFERENCES

[1] Abdallah, T. (1982),"Application de la Recherche Opérationnelle et de la Micro-Informatique aux Problèmes Miniers",Thesis of Dr. Ing.,E.N.S.M.I.M, Nancy, France.

[2] Manula C.B, E.K. Albert and R.V. Ramani,(1976)," Application of a Total System Surface Mine Simulator to Coal Stripping", Vol.II, Pennsylvania State University.

[3] O'Neill, T.J., (1966),"Computer Simulation of Materials Handling in Open pit Mining",M.Sc. Thesis, Pennsylvannia State University.

[4] S.M.E., (1973),"Systems Engineering", Vol.2, Chapter 30, pp. 27-30.

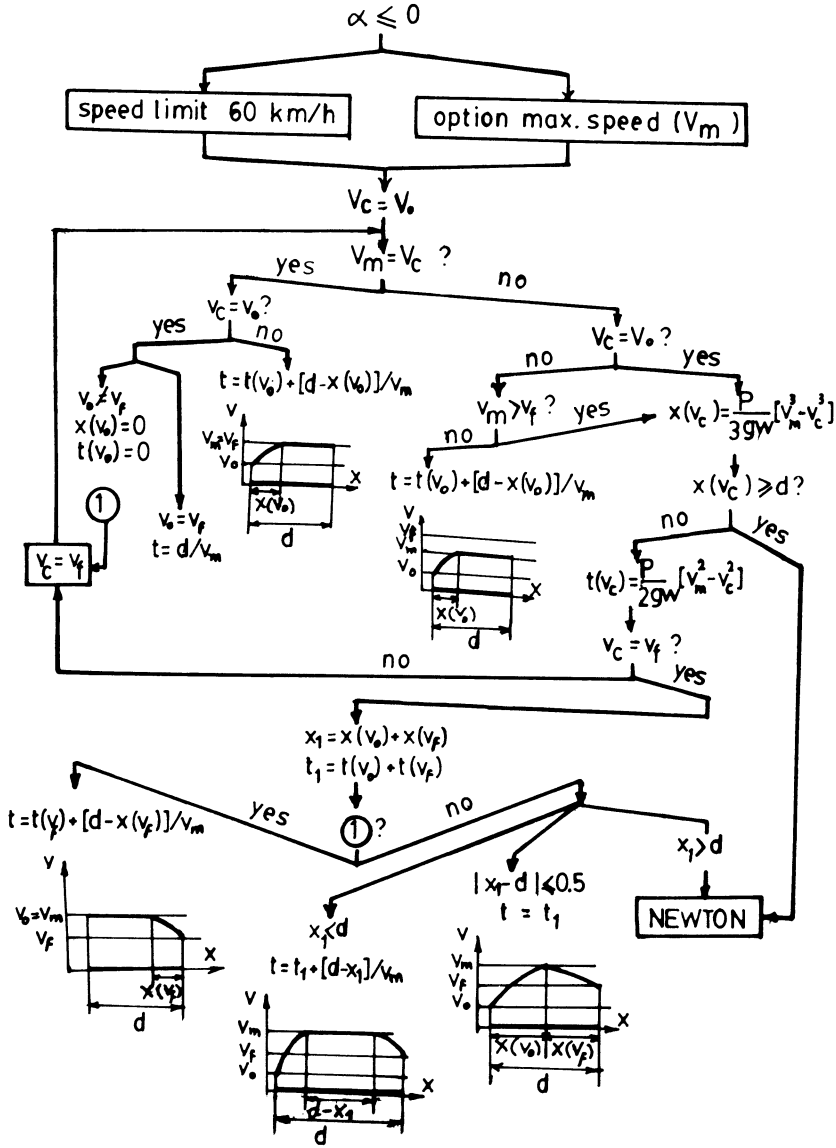


Figure 3 : Schema of travel time calculation for flat or downhill grade.

APPENDIX 1

The Calculation of $t(V_\ell)$ and $X(V_\ell)$

Vehicle motion around an initial point X_o can be expressed by Taylor's expansion.

$$X = X_o + \dot{X}_o t + \ddot{X}_o \frac{t^2}{2} \tag{1.1}$$

$$\dot{X} = \dot{X}_o + \ddot{X}_o t \tag{1.2}$$

Where X_o, $\dot{X}_o$ and $\ddot{X}_o$ represent the initial state. $\dot{X}_o$ and $\ddot{X}_o$ are the vectors of speed and acceleration respectively.

Equations (1.1) and (1.2) give :

$$\dot{X}^2 = \dot{X}_o^2 + 2\ddot{X}_o \cdot \Delta X \tag{1.3}$$

where $\Delta X = X - X_o$

But equations (7) give :

$$X = t.\, V_\ell - \frac{1}{2\alpha g}(V^2 - V_c^2) \tag{1.4}$$

Let $V = V_\ell$ we obtain :

$$V_\ell^2 = V_c^2 + 2\alpha g.\ \Delta X \tag{1.5}$$

where $\Delta X = t.V_\ell - X$

Compare equations (1.3) and (1.5), then :

$$\ddot{X}_o = \alpha g \tag{1.6}$$

Equation (1.2) can now be written as :

$$V_\ell = V_c + \alpha g\ t(V_\ell)$$

or $$t(V_\ell) = (V_\ell - V_c)/\alpha g \tag{1.7}$$

Substituting t in equation(1.4)by equation (1.7), we obtain :

$$X(V_\ell) = \frac{t(V_\ell)}{2} \cdot (V_\ell - V_c) \tag{1.8}$$

APPENDIX 2

CALCUL : Program Listing.

```
10  DIM D(2,50),PEN(2,50),V0(2,50),TT(10,10,2),DEN(2),KT$(10),VK(2)
12  HOME : VTAB 10: PRINT "CALCUL DU TEMPS DES TRAJETS": VTAB 20: PRINT "P
    OUR CONTINUER TAPEZ OUI": INPUT A$: IF A$ = "OUI" THEN 15
14  GOTO 2000
15  HOME :DD$ =  CHR$ (4)
20  PRINT DD$;"OPEN DENSITE": PRINT DD$;"READ DENSITE"
30  FOR I = 1 TO 2: INPUT DEN(I): NEXT
40  PRINT DD$;"CLOSE DENSITE"
50  PRINT DD$;"OPEN RERO": PRINT DD$;;"READ RERO"
60  FOR I = 1 TO 5: INPUT R(I): NEXT
70  PRINT DD$;;"CLOSE RERO"
80  PRINT DD$;"OPEN TYPE": PRINT DD$;"READ TYPE"
90  INPUT TC%: FOR I = 1 TO TC%: INPUT KT$(I): NEXT
100  PRINT DD$;"CLOSE TYPE"
110  PRINT DD$;"OPEN TRAJET": PRINT DD$;"READ TRAJET"
120  INPUT NT: FOR I = 1 TO NT: INPUT A1$(I): INPUT B1$(I): INPUT MA(I): NEXT

130  PRINT DD$;"CLOSE TRAJET"
140  INPUT "COEFFICIENT DE PUISSANCE?(%) ";TAU: PRINT :VA = 0
142  INPUT "TEMPS REEL D'UN POSTE?(HR.-MIN.) ";A$: PRINT
144 A =  LEN (A$): FOR I = 1 TO A: IF  MID$ (A$,I,1) = "-" THEN 146
145  NEXT
146 B =  VAL ( LEFT$ (A$,I - 1)):C =  VAL ( RIGHT$ (A$,A - I)):TP0ST = 50 *
    (B + C / 60)
150  PRINT "VOULEZ-VOUS IMPOSER UNE VITESSE MAXI.": INPUT "?(O/N) ";V$: PRINT
    : IF V$ = "N" THEN 170
160  INPUT "VMAX.?(KM/HR) ";VIT: PRINT
170  PRINT DD$;"OPEN VIT": PRINT DD$;"WRITE VIT": PRINT TP0ST: PRINT TAU: PRINT
    V$: PRINT VA
172  IF V$ = "O" THEN  PRINT VIT
174  PRINT DD$;"CLOSE VIT"
175  IF V$ = "O" THEN VM = VIT * 10 / 36
180  IF VA = 1 THEN 242
200  FOR J = 1 TO TC%
210  PRINT DD$;"OPEN";KT$(J): PRINT DD$;"READ";KT$(J)
220  INPUT M2A(J): INPUT W1(J): INPUT CV(J): INPUT BC(J)
230  PRINT DD$;"CLOSE";KT$(J)
240  NEXT J
242  IF NT = 1 THEN I = NT: GOTO 250
245  FOR I = 1 TO NT
250 T$ = "T" + A1$(I) + "-" + B1$(I)
260  PRINT DD$;"OPEN";T$: PRINT DD$;"READ";T$
270  FOR M = 1 TO 2
280  IF M = 1 THEN  INPUT S1%:S = S1%: GOTO 300
290  INPUT S2%:S = S2%
300  FOR J = 1 TO S: INPUT D(M,J): INPUT PEN(M,J): INPUT V0(M,J): NEXT J: NEXT
    M
305  INPUT SA: INPUT SR
310  PRINT DD$;"CLOSE";T$
312 A$ = A1$(I) + "-" + B1$(I)
315  PRINT : PRINT "DEBUT DU CALCUL SUR TRAJET:": PRINT : PRINT  SPC( 10);
    A$: PRINT
320  FOR I1 = 1 TO 5:R = R(I1):VB = 0
330  FOR LL = 1 TO TC%:N = M2A(LL)
340  IF MA(I) = N OR N = 3 THEN 360
350  FOR MM = 1 TO 2:TT(I,LL,MM) = 0: NEXT : GOTO 1425
360 N = MA(I):P1 = DEN(N) * BC(LL):W0 = (75 / 1000) * W1(LL) * TAU / 100
480  FOR M = 1 TO 2
490  IF M = 1 THEN S = S1%:CHARGE = P1 + CV(LL):SS = SA: GOTO 510
500 S = S2%:CHARGE = CV(LL):SS = SR
510 TEMPS = 0
512  FOR L = 1 TO S: IF L = 1 THEN V0 = 0: GOTO 515
514 V0 = VV
515 RT = (R + PEN(M,L)) / 100:VV = V0(M,L) * 10 / 36
540  IF RT (  = 0.0 THEN 812
550 VL = W0 / (RT * CHARGE):G1R = RT * 9.81:G2R = 1 / (2 * G1R):A = 0
560  IF V$ = "N" THEN VM = 50 / 3
570  IF VL ) VM THEN 772
572  IF V0 (  ) VL THEN 580
574  IF V0 = VV THEN T = D(M,L) / VL: GOTO 960
576 TVL = 0:XVL = 0: GOTO 630
580 V =  ABS (VL - V0):TVL = V / G1R
```

CALCUL : Program Listing.

```
590 XVL = (TVL / 2) * V
595  IF V0 <  > VV THEN 610
600  IF XVL <  = D(M,L) / 2 AND V0 < VL THEN 630
605  IF XVL <  = D(M,L) / 2 AND V0 > VL THEN 640
607  GOTO 672
610  IF XVL < D(M,L) THEN A = 1: GOTO 650
620  GOTO 672
630 T = 2 * TVL + (D(M,L) - 2 * XVL) / VL: GOTO 960
640 T = TVL + (D(M,L) - XVL) / VL:VV = VL: GOTO 960
650  IF VV =  > VL THEN 640
660 V = VL - VV:T1 = V / G:R:X1 = (T1 / 2) * V
662  IF A <  > 1 THEN 670
665  IF  ABS (D(M,L) - XVL - X1) <  = .5 THEN T = TVL + T1: GOTO 960
668  IF D(M,L) < XVL + X1 THEN 672
670 T = TVL + T1 + (D(M,L) - XVL - X1) / VL: GOTO 960
672 V1 = VL
673  GOSUB 5500
676 D = D(M,L):A1 = 2 * VL / G1R:A3 = A1 / 2:A2 = VL * A3:A5 = 1 / G1R:A6 =
    G2R * (V0 * V0 + VV * VV):A4 = A3 * (V0 + VV) + A2 *  LOG ((VL - V0) *
    (VL - VV)) + A6
678  DEF  FN F(V) =  - D - A5 * V * V - A1 * V - A2 *  LOG ((VL - V) * (VL
     - V)) + A4
680  DEF  FN F1(V) =  - A1 + 2 * A2 / (VL - V) - 2 * A5 * V: GOSUB 8000
682 A =  FN F(V)
683 B = A5 * V * V - A6
684 T = (A + D + B) / VL: GOTO 960
772 VN = V0:AB = 1:A = 0:D = D(M,L)
774  IF VM > VN THEN 788
776  IF VM < VN THEN 786
778  IF V0 = VV THEN T = D / VM: GOTO 960
780  IF AB = 1 THEN A = 1:X1 = 0:T1 = 0: GOTO 803
782  GOTO 810
786 VV = VM: GOTO 810
788 LV =  LOG ((VL - VM) / (VL - VN))
790 T1 = ( - VM + VN - LV * VL) / G1R:X1 = VL * T1 - G2R * (VM * VM - VN *
    VN)
792  IF A = 1 THEN T2 = T1:X2 = X1: GOTO 810
798  IF AB = 2 THEN 804
800  IF X1 <  = D / 2 AND V0 = VV THEN T2 = 2 * T1:X2 = 2 * X1: GOTO 810
801  IF X1 < D AND V0 <  > VV THEN 803
802 V1 = VM: GOTO 673
803 AB = 2:X2 = X1:T2 = T1:VN = VV: GOTO 774
804 X2 = X2 + X1:T2 = T2 + T1: IF X2 =  < D THEN 810
806  IF  ABS (X2 - D) <  = .5 THEN T = T2: GOTO 960
808 V1 = VM: GOTO 673
810 T = T2 + (D - X2) / VM: GOTO 960
812 WV = CHARGE / (9.81 * W0):W3V = WV / 3:W2V = WV / 2
813  IF V$ = "N" THEN VM = 50 / 3
815 VL = VM
826 VN = V0:AB = 1:A = 0
828  IF VM <  > VN THEN 833
830  IF V0 = VV THEN T = D(M,L) / VM: GOTO 960
831  IF AB = 1 THEN A = 1:T1 = 0:X1 = 0: GOTO 860
832  GOTO 870
833  IF AB = 1 THEN 836
834  IF VM <  = VN THEN VV = VM: GOTO 870
836 XVM = W3V * (VM ^ 3 - VN ^ 3): IF XVM >  = D(M,L) THEN 890
838 TVM = W2V * (VM * VM - VN * VN)
840  IF A = 1 THEN X1 = XVM:T1 = TVM: GOTO 870
845  IF AB = 2 THEN 865
850 X1 = XVM:T1 = TVM: IF V0 = VV THEN D = D(M,L) / 2:FI = .25: GOTO 875
860 AB = 2:VN = VV:D = D(M,L):FI = .5: GOTO 828
865 X1 = X1 + XVM:T1 = T1 + TVM: GOTO 875
870 T = T1 + (D - X1) / VM: GOTO 960
875  IF  ABS (X1 - D) <  = FI THEN 884
880  IF X1 > D THEN 890
882 T1 = T1 + (D - X1) / VM
884  IF V0 = VV THEN T = 2 * T1: GOTO 960
886 T = T1: GOTO 960
890 D = D(M,L):A1 = W3V * (V0 ^ 3 + VV ^ 3)
891  DEF  FN G(V) =  - D + 2 * W3V * V ^ 3 - A1
892  DEF  FN G1(V) = 2 * WV * V * V
```

CALCUL : Program Listing.

```
893 V1 = VL: GOSUB 5500: GOSUB 8000
895 A =  FN G1(V) / 2
896 T = A - W2V * (V0 * V0 + VV * VV)
960 T = T / 60
970 TEMPS = TEMPS + T
972  NEXT L
1410 TT(I,LL,M) = TEMPS
1420  NEXT M:VB = VB + 1
1425  NEXT LL
1426 A = 0: FOR M = 1 TO 2: FOR J = 1 TO TC%:A = A + TT(I,J,M): NEXT J: NEXT
     M
1427 VK = (SA + SR) * VB / A:VK = VK * 3 / 50:VK = ( INT (VK * 10)) / 10
1430  IF V$ = "O" THEN A$ = "VM" +  STR$ (VIT): GOTO 1445
1440 A$ = "VL"
1445 A$ = "T" +  STR$ (I) + A$
1446  IF I1 (  ) 1 THEN 1450
1447  PRINT DD$;"OPEN";A$: PRINT DD$;"DELETE";A$
1450  PRINT DD$;"OPEN";A$: PRINT DD$;"APPEND";A$: PRINT DD$;"WRITE";A$
1460  FOR J = 1 TO TC%: FOR J1 = 1 TO 2: PRINT TT(I,J,J1)
1465  NEXT J1: NEXT J
1470  PRINT DD$;"CLOSE";A$
1475  IF V$ = "O" THEN A$ = "VM": GOTO 1480
1477 A$ = "VL"
1480  IF I = 1 AND I1 = 1 THEN 1484
1482  GOTO 1488
1484  PRINT DD$;"OPEN";A$: PRINT DD$;"DELETE";A$
1488  PRINT DD$;"OPEN";A$: PRINT DD$;"APPEND";A$: PRINT DD$;"WRITE";A$
1490  PRINT VK
1492  PRINT DD$;"CLOSE";A$
1493 A$ = A1$(I) + "-" + B1$(I)
1494  HOME : PRINT : PRINT "TRAJET = ";A$: PRINT : PRINT "RESIST. AU ROULT
    . = ";R; %": PRINT : PRINT "VIT. MOYENNE = ";VK;" KM/HR": PRINT
1500  NEXT I1: IF NT = 1 THEN 1510
1505  NEXT I

1510  IF VA = 1 THEN 1900
1520  IF V$ = "O" THEN 1540
1530  PRINT "VOULEZ-VOUS IMPOSER UNE VITESSE MAXI.": GOTO 1550
1540  PRINT "CALCUL BASE SUR UNE VIT.LIMITE"
1550  INPUT "?(O/N) ";A$: PRINT
1560  IF A$ = "N" THEN 1900
1570  IF V$ = "O" THEN VA = 1: HOME : GOTO 170
1580 V$ = A$:VA = 1: HOME : GOTO 160
1900  PRINT : PRINT : PRINT  SPC( 13);"FIN DU CALCUL"
2000  END
5500  IF V0 ) VV THEN VX = V0: GOTO 5520
5510 VX = VV
5520 V = (V1 + VX) / 2:V2 = VX
5530  RETURN
8000 A = V
8005 B = A
8008  IF RT ) 0 THEN 8023
8015 C1 =  FN G1(B):C =  FN G(B): GOTO 8040
8023  IF B =  ) VL THEN B = VL - .1
8025 C1 =  FN F1(B):C =  FN F(B)
8040 A = B - C / C1
8050  IF  ABS (C) ) .5 THEN 8005
8060 V = A:A = 3
8070  RETURN
```

"CAPS: A SIMULATOR OF SHOVEL-TRUCK SYSTEM IN OPEN PIT MINES USING A MICRO-COMPUTER"

BY

T.Abdallah[1] and C. Chambon[2]

1) Dr. Ing., Department of Mining, Petroleum and Metallurgical Engineering, Faculty of Eng., Cairo University, GIZA (Egypt).
2) Professor, Ecole Nationale Supérieure de la Métallurgie et de l'Industrie des Mines, Parc de Saurupt, 54042 Nancy (France).

INTRODUCTION

The model presented here is meant to minimize management's risk and supplying open pit miners with a scientific tool which aids in decision making. We have tried to use the advantages offered by micro-computers and especially the interactive dialogue programation and the visualization of results. A model called CAPS (in french: CAmions-Pelle Système) simulates a shovel-truck system. The program, written for Apple II 48 K, is compiled by the package TASC to reduce the calculation time which does not exceed 30 seconds for 10 trucks assigned to a given shovel. CAPS works in connection with two other programs which are MENU and CALCUL. The first prepares all data files necessary for the simulation, while the second calculates, using this information, truck travel time. Three fundamental differences distinguish CAPS from previous simulators. First, the decision is still made by the user and not the computer; this is possible due to the interactive dialogue programation. The second is that previous simulators usually describe the empirical distribution of loading and dumping times by a standard probability distribution. Although this frees a sizeable block of computer storage the user, and especially the miner, is obliged to sample many observations and to have a certain knowledge of statistics to apply a fitness test.

For this reason, CAPS generates these times directly from the empirical distribution. The last difference is the deterministic simulation of truck movement, which is based on the curves of constant power [1] and not the characteristic curves supplied by the manufacturers. CAPS and MENU are presented in the paper.

DESCRIPTION OF MENU

The program prepares data files required for both the calculation of truck travel time (CALCUL) and the simulator CAPS. We have tried to use the same terminology employed by the miners. MENU offers seven options which are the following [2]:

1- Definition of Road Coordinates

A maximum of 10 roads, each of 50 elements (one way) can be defined. A road is marked by the bench level of both loading and dumping points and by the hauled material. Every element is defined by a length (meters), a grade (%), and a curve speed (Km/h) imposed at the end of the element.

2- Truck Characteristics

The different type (10 maximum) of trucks are mentioned. Each type is defined by the nominal power (hp or ch), unladen truck weight (tons), capacity (cubic meters) and the material being hauled by this type, if the user wishes to assigne a certain type for the loading of waste only, for example.

3- Histograms

This enables the user to introduce into a file the samples of either loading or dumping observed times. Then a histogram representing the distribution of these times is drawn (figure 1) and is stored in a file.

4- Rolling Resistance

Five values (%) are defined. They take into consideration both the maintenance of the road and the expected changes in the weather. These values are necessary for the calculation of truck travel time[3].

5- Open Files

Necessary for options 1 and 3.

6- The Density

Defined (in gm/cm^3) for ore and waste.

7- Numbers of Trucks

The type and the number of truck are mentioned in order to enable the micro-computer to distinguish one truck from another. As: CATER. 777- N°.113.

```
*************************************
*                MENU               *
*                ----               *
*                                   *
* (1) DEFINIR LES COORDONNEES       *
*     D'UN TRAJET                   *
*                                   *
* (2) DEFINIR LES CARACTERISTIQUES  *
*     DES CAMIONS                   *
*                                   *
* (3) CHRONOMETRAGE                 *
*                                   *
* (4) DEFINIR LA RESISTANCE AU      *
*     ROULEMENT                     *
*                                   *
* (5) CREATION D'UN FICHIER         *
*                                   *
* (6) DEFINIR LA DENSITE EN VRAC    *
*                                   *
* (7) LA NUMEROTATION DES CAMIONS   *
*************************************
TAPEZ LE NUMERO ENTRE () 3
AVEZ-VOUS DEJA DEFINI LES

CARACTERISTIQUES DES CAMIONS?(O/N) O
INTRODUIRE UN NOUVEAU CHRONOMETRAGE?
(O/N) N

TRACER UN HISTOGRAMME

TYPE D'OPERATION:
                (1) CHARGEMENT
                (2) DECHARGEMENT

TAPEZ LE NUMERO ENTRE () 1

MATERIAU :
          (1) STERILS
          (2) MINERAI

TAPEZ LE NUMERO ENTRE () 2

CHARGE PAR LA PELLE: RICHIER

TYPE DU CAMION: CATER.773A

  FREQ.                        CHARGEMENT
   1 +                         DU MINERAI
     I                         CAMION :
     I                         CATER.773A
     I                         PELLE :
  .8+                          RICHIER
     I
     I
     I
  .6+
     I
     I
     I
  .4+
     I          III
     I          IIIIII
     I          IIIIII
  .2+           IIIIII
     I       IIIIIIIIIII
     I       IIIIIIIIIII
     I       IIIIIIIIIIIII
  0.0+--/----+----+----+--+-- TEMPS(MINS)
              5.5   6     6.5
VOULEZ-VOUS RETOURNEZ AU MENU?(O/N) N
```

Figure 1: Listing of Printout from MENU program.

DESCRIPTION OF CAPS

Every shovel in an open pit mine is considered as an independent system. Figure 2 shows an open pit truck-shovel system as it is shematized into CAPS. For one shovel, a number of trucks (10 maximum) is assigned. Three types of materials can be extracted from one loading point. These include ore, waste, and Ore & waste combined. In the last case, the percent of ore to the total material must be defined. When trucks are loaded they are oriented towards the proper discharge point according to the type of material hauled (ore or waste). While travel time is already calculated by CALCUL and stored in files, the loading and dumping times, which are considered as random variables, are generated by the Monte-Carlo Method. This is based on the empirical distribution of these variables. The histograms representing these distributions are also already known and stored in files. This is done by option 3 of MENU. Results of a shift simulation give the production (cubic meters) of each truck and the waiting times (minutes),[2,4].

EXAMPLE

As data storage is done by MENU, the travel time of each type of truck is computed by CALCUL for a given density, (ore and waste), using 5 values of the rolling resistance. When CAPS is executed the values of the density for either waste (2.1 gm/cm^3) or ore (1.7 gm/cm^3) are stated (Figure 3a). If these values are not valied a message asking the user to define again the new density (suing MENU-Option 6) and then to compute again the travel time. But if these values are still valid, a shovel type is chosen which is, in the example, a caterpillar 992 B. The shovel will extract the waste (material No.1) and is assigned to a loading point (level 265.64). Let the rolling resistance be 7%. Once the number of trucks assigned to the shovel is mentioned (3 in the example) the number of each one is stated. Here, the micro-computer identifies the type of each truck. In the given example, 2 trucks (No. 121 and No. 122) are of type caterpillar 773B and the third one (No. 131) is a cater pillar 777. The identification is done by means of option7 MENU. Consequently, the micro-computer can read files of these types of trucks containing times of both dumping and loading by the shovel 992 B. Then, the second configuration of travel time calculation appears on the screen. This is the constraint on the maximum speed of trucks with the corresponding average speed. CAPS calculates the production of each truck (in cubic meters) and the waiting times (in minutes) of shovel and trucks.

In this example, no waiting times are observed at dumping or on the haul road (Figure 3b). Then, the user can repeat the simulation using the other option of speed. If not, he can

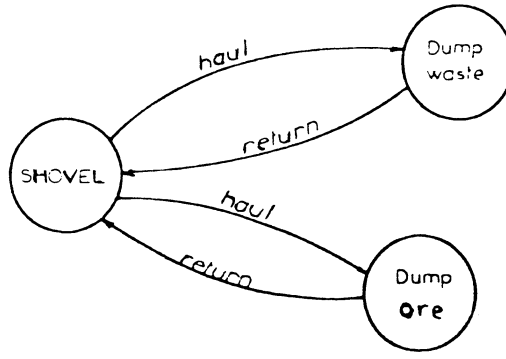


Figure 2: An open pit truck-shovel system as it is Schematized into CAPS.

-a-

```
        SIMULATION D'UNE EXPLOITATION

                A CIEL-OUVERT

          ( SYSTEME : PELLE-CAMION )
POUR CONTINUER TAPEZ OUI
?OUI
LA DENSITE EN VRAC (GM./CM.3)
                DU STERILS = 2.1
                DU MINERAI = 1.7

ETES-VOUS D'ACCORD?(O/N) O

TYPE DE PELLE : 992B

MATERIAU EXTRAIT :
                    (1)STERILS
                    (2)MINERAI
                    (3)MINERAI&STERILS

TAPEZ LE NUMERO ENTRE () 1

POINTS DE CHARGEMENT DU STERILS

      (1) 265.64
TAPEZ LE NUMERO ENTRE () SELON LE POINT
CHOISI   ?1

VALEURS DE LA RESISTANCE AU ROULEMENT:
                                (1) 6  %
                                (2) 7  %
                                (3) 8  %
                                (4) 9  %
                                (5) 10 %

TAPEZ LE NUMERO ENTRE () SELON
LA RESISTANCE CHOISIE   ?2

NBRE DES CAMIONS AFFECTES(MAX.10) : 3

  NB. DU 1 ER CAMION   : ?121

  NB. DU 2 EME CAMION  : ?122

  NB. DU 3 EME CAMION  : ?113

TEMPS DU TRAJET CALCULE A PARTIR D'UNE:

  (1)VIT.LIMITE

     (VIT.MOYENNE= 18.3 KM/HR)

  (2)VIT.MAXI.= 20 KM/HR

     (VIT.MOYENNE= 15.9 KM/HR)

TAPEZ LE NUMERO ENTRE () 1

                CALCUL
            PELLE : 992B
+-----------------------------------+
*CAMION* CYCLES *     MT.3      *
*       *-----*------*-------*
* NB.   *MIN.*STR.* MIN. * STR. *
+-----------------------------------+
* 121  *  8  *  38 *  8  *   722  *
* 122  *  8  *  38 *  8  *   722  *
* 113  *  8  *  37 *  8  *  1184  *
+-----------------------------------+
          LA SOMME = 8       2628
```

-b-

```
POUR CONTINUER APPUYER SUR
N'IMPORTE QUEL BOUTON
     LE BILAN D'ATTENTE(EN MINUTES)

      PELLE : 55

            LES CAMIONS
      +-----------------------+
      * POINT  *STERILS*MINERAI*
      +-----------------------+
      * CHARG. :     127.2    *
      *-----------------------*
      * DECHR. : 8    : 8     *
      *-----------------------*
      *        A: 8    : 8     *
      * TRAJET :      :       *
      *        R: 8    : 8     *
      *-----------------------*
      *LA SOMME:     127.2    *
      +-----------------------+
POUR CONTINUER APPUYER SUR
N'IMPORTE QUEL BOUTON
REFAIRE LE CALCUL AVEC VIT. MAXI.=

20 KM/HR ?
     (VIT.MOYENNE= 15.9 KM/HR)

(O/N) O
              CALCUL
          PELLE : 992B
+-----------------------------------+
*CAMION* CYCLES *     MT.3      *
*       *-----*------*-------*
* NB.   *MIN.*STR.* MIN. * STR. *
+-----------------------------------+
* 121  *  8  *  35 *  8  *   665  *
* 122  *  8  *  35 *  8  *   665  *
* 113  *  8  *  34 *  8  *  1088  *
+-----------------------------------+
          LA SOMME = 8       2418

POUR CONTINUER APPUYER SUR
N'IMPORTE QUEL BOUTON
     LE BILAN D'ATTENTE(EN MINUTES)

      PELLE : 77.1

            LES CAMIONS
      +-----------------------+
      * POINT  *STERILS*MINERAI*
      +-----------------------+
      * CHARG. :     115.7    *
      *-----------------------*
      * DECHR. : 8    : 8     *
      *-----------------------*
      *        A: 8    : 8     *
      * TRAJET :      :       *
      *        R: 8    : 8     *
      *-----------------------*
      *LA SOMME:     115.7    *
      +-----------------------+
POUR CONTINUER APPUYER SUR
N'IMPORTE QUEL BOUTON
REFAIRE LE CALCUL AVEC VIT. LIMITE ?
     (VIT.MOYENNE= 18.3 KM/HR)

(O/N) N
CHANGEMENT D'AFFECTATION DES CAMIONS A

LA PELLE 992B ?(O/N) N

CHANGEMENT DE LA RESIST. AU ROULEMENT?

(O/N) N

CHANGEMENT DE PELLE OU DU TRAJET?

(O/N) N
```

Figure 3: Listing of Printout from CAPS Program.

either reassign trucks or change the rolling resistance to take into consideration weather changes or maintenance conditions of the road. He can also change the shovel type or assign the shovel to another loading point. The listings of the programs are enclosed in appendices 1 and 2.

CONCLUSIONS

CAPS represents an effective support in decision making, as it can treat many typical problems of daily occurrence in open pit mines, such as truck assignements, choice of a dumping point and both the design and the maintenance of road profiles. Real situations at Decazeville surface coal mine in France have been chosen to test CAPS. A strong correlation between the number of truck cycles calculated and those realized have been found. Moreover, it is sufficient to simulate only one shift to obtain a precision of $\pm$ 0.5 on the number of cycles, and the corresponding precisions on the waiting times are also sufficient.

REFERENCES

[1] Abdallah, T. and C. Chambon, (1982), "Micro-Computers: A New Simulation Program for Truck Travel-time in Open Pit Mines", paper Submitted to this Congress.

[2] Abdallah, T., (1982), "Application de la Recherche Opérationnelle et de la Mico- Informatique aux Problèmes Miniers", Thesis of Dr.Ing.,E.N.S.M.I.M., Nancy, France.

[3] Caterpillar Tractor Co.,(1976), "Le Terrassement",

[4] Peurifoy, R.L., (1970), "Construction Planning Equipment and Methods", McGraw-Hill, pp. 281-298.

Program listing available from authors.

CHEMICAL PROCESS CALCULATIONS USING THE MINI*DES PROGRAMS

JohnH.Erbar (1), Mike W. Slavin (2), Martin A. Branton (3)

1) School of Chemical Engineering, Oklahoma State University, Stillwater, Oklahoma, USA 74078.
2) Managing Director, M.O.M. (Offshore) Ltd., Gravesend, Kent
3) Senior Engineer, M.O.M. (Offshore) Ltd., Gravesend, Kent

Introduction

The advent of low priced microcomputers has opened a new vista for the chemical process design engineer. No longer is he completely tied to large mainframe computers under the autocratic domination of the EDP group. He can now use microcomputers to do preliminary process design, investigate the effects of changes in process conditions, characterize feeds, etc. Many of the traditional unit operations can be quickly, easily and economically simulated using the MINI*DES family of programs on microcomputers. Certainly, these simulations require more time to get a solution than would be required on the large mainframes, but the incremental time requirements are not excessive. In some cases, (tray by tray calculations) the final quality of the microcomputer solution may be differentially less than that obtained from the main frame, but these deficiencies are not usually very severe.

In this presentation, we will describe the current MINI*DES family of programs and discuss some of our future developments to be added to the suite of programs. In addition, selected features of

MINI*DES programs will be illustrated to demonstrate the power of the programs.

The MINI*DES Programs

All of the programs in the MINI*DES family are designed to operate on 64K microcomputers and take full advantage of the interactive capabilities of this class of computer. These interactive features permit the user to alter the program input data either one or several variables at a time to evaluate the effects of input parameters. Another distinct advantage (we believe) of the interactive capabilities is

that the process engineer is intimately involved in the process design calculations and not just a passive by-stander as is the case when using the large scale batch simulator. Our clients have observed that these programs have an excellent side benefit in the better development of the thinking/analytical processes of the engineers who use MINI*DES.

The following discussion provides an overview of each of the programs in the MINI*DES family. These programs can broadly be categorized as "support" programs or "design" programs. The support programs are designed to process selected data and either produce output files to be used in the "design" programs or accept output files from the "design" programs and produce final output reports. The "design" programs do the process unit operation calculations.

MINI*SIM

The MINI*SIM program is the workhorse of the MINI*DES family. This program is capable of performing 13 different equilibrium calculations (Table I) along with compressor calculations and shortcut distillation and pipeline flow (single and two phase) calculations. In addition, MINI*SIM can be used to perform RVP and EFV calculations, predict CO_2 freezeout, estimate the heating value of a gas and adjust the critical properties of C_6+ type fractions. (An example problem demonstrating the necessity for this latter capability will be given later.) MINI*SIM gives the user the capability of modifying and saving streams, modifying the input parameters, changing the component identification list, adding streams together, etc. All of these calculational features are initiated by two character mnemonics (Table 2). The program can handle a maximum of 25 components and retain between 10 and 45 streams in working memory (depending on the number of components in the system).

The Soave Redlich Kwong (SRK) (12) equation of state is used to predict the thermodynamic properties of the fluids, thus MINI*SIM is by and large restricted to hydrocarbon and acid gas systems. Liquid densities are estimated by the Hankinson-Thomson COSTALD method. Up to 19 C_6+ fractions may be considered in a single problem. C_6+ fraction properties are estimated by a modified GPA RR 13 (6) procedure. The temperature/pressure limits of the SRK equation of state are difficult to assess; composition, C_6+ characterization, component identity, etc., play a very significant role in defining these limits. We believe, however, that the SRK can be used reliably over the temperature range -160 C to 310 C and pressures up to 700 bars, provided any C_6+ fractions have been "properly" characterized and no multiple phase behavior (a solid phase or an extra liquid phase) is encountered. K

values for the light components (N_2, C_1, C_2) are usually estimated to within about 2-8%; the intermediate component (C_3, C_4's, C_5's, C_6's) K value will be estimated within about 5% and the heavier component K values will usually be estimated to within 15-20%. Enthalpies will usually be estimated to within 4 kJ/kg. Vapour densities are estimated to within 4% except in the critical region where the errors may be as high as 15%. Liquid phase densities are estimated to within about 4% except when the C_6+ fraction is very heavy (500°C NBPT and higher) and in the region of the critical point of the mixture. Thus, the thermodynamics property prediction procedure are more than adequate for most process designs.

MINI*SIM has been used to design/model a number of diverse processes. These processes include:

(1) a cascade refrigeration system for LNG refrigeration
(2) turbo expander plants for high C_2 recovery
(3) a heavy oil gas lift process
(4) oil recovery processes for platforms
(5) high pressure gas dew point control/liquid recovery processes for a platform
(6) high pressure gas transmission lines with single and two phase flow
(7) super critical gas transmission lines
(8) a preliminary model of the Ryan-Holmes process for CO_2 recovery
(9) mixed refrigerant gas liquids recovery processes
(10) receiving terminal systems with a variety of process of objectives
(11) crude oil staged flash pressure profile optimization, and
(12) An olefin storage tank refrigeration loop.

These processes represent a few examples of the applications where MINI*SIM has been used industrially and such a variety of applications adequately demonstrates the power of MINI*SIM. We believe that such applications are limited only by the users imagination.

To demonstrate the technical power of MINI*SIM we have developed a C_6+ characterization for an almost typical "crude oil" system. The basic information available for this problem is given in Table 3. The C_6+ TBP data were manipulated by C6PLUS (a program to be described later) to produce the molecular weight, specific gravity and boiling point data shown in Figure 1. The predicted phase behavior of the system without and with C_6+ critical property adjustment (14) is shown in Figure 2. Clearly, if we are to predict properly the phase behaviour of naturally occurring

systems, C_6+ fraction critical properties are essential. The K values predicted on the basis of the adjusted C_6+ critical properties are compared with the experimental values in Figure 3. The agreement between the predicted and experimental K values for C_1 through C_4's is certainly good but not excellent. The agreement for the C_5's and heavier is, at worst, acceptable. However, considering the experimental problems extant at the time the work was performed, this quality of agreement is the best that can be expected. Under any circumstance however, we believe that the overall quality of agreement between predicted and experimental value is, at least, adequate and is certainly better than basing any design on unadjusted critical properties of the C_6+ fractions. Similar effects can be shown for "condensate" type systems.

Note: In this exercise, we have made no attempt to reproduce the K value data, our sole objective being to match the available fluid phase behaviour, i.e., liquid bubble point and the vapour formation at the various pressures.

To summarize, we believe MINI*SIM to be a program that can be used in a broad variety of process analysis problems that occur in design and operations. The thermodynamic property prediction procedures are reliable and can provide extremely good representation of hydrocarbon systems if the feed characterization is properly handled.

C6PLUS

C6PLUS is a program designed to convert a partial TBP analysis boiling (a true boiling point distillation with a whole fraction specific gravity and molecular weight) to a complete TBP analysis (a true boiling point analysis with several subfraction with the molecular weight and specific gravity of each subfraction estimated). The programs can also convert an ASTM D158 (or equivalent) or a "chromatographic TBP" to a partial TBP and thence to a complete TBP analysis. Our experience has repeatedly shown that the C_6+, C_7+ type fractions must be broken into in several subfractions if one expects to make reasonable phase behaviour predictions for the petroleum/natural gas systems (particularly those in the North Sea). You should note that this program should not be used if you have a complete TBP analysis available; C6PLUS is to be used to estimate a complete TBP analysis from less satisfactory data (such as an ASTM distillation).

The program functions in the following fashion:

(1) if a non-TBP distillation is specified, that distillation is converted to a TBP distillation using the appropriate

empirical conversion routine,

(2) the TBP distillation is broken into a user designated number of fractions and the mid boiling point of each fraction is estimated,

(3) the specific gravity of each fraction is initially estimated based on the UOP K of the total fraction, using some empirical adjustments,

(4) the estimated specific gravities of each cut are adjusted until the calculated specific gravity of the total fraction matches the input specific gravity, and

(5) the molecular weight of each cut is estimated using the Lee Kesler equation. These molecular weights are then adjusted to match the user entered molecular weight of the total fraction.

While C6PLUS is based on considerable empiricism and experience, it does provide a set of C_6+ subfractions specifications that are reasonably consistent with the available experimental data and that are completely in volume-mass-mol balance (3). Figure 1 shows the ability of C6PLUS to estimate qualitively complete TBP analyses.

C6PLUS has a limited mnenonic command structure which allow selected changes in the input data, and different computational methods to be requested at selected points in the program. An output data file can be saved which serves as input to MINI*SIM to avoid the tedium of entering the appropriate data for each C_6+ subfraction.

HANDM

The HANDM program accepts a MINI*SIM output file and produces a mol balance table suitable for inclusion in reports. The program has very limited computational capabilities; mainly those used to estimate pseudo critical properties, perform units conversion, calculate volumetric flow rates, etc. The output structure and contents of HANDM can be altered to suit the individual user's needs.

MINI*TXT

MINI*TXT is a program for performing tray by tray distillation calculations. The program is capable of handling between 2 and 25 components, from 35 to 148 trays (depending on the number of components), up to four feeds, zero to four side draws (either vapour or liquid) and zero to four interstage coolers/reboilers. Thermodynamic properties are estimated from polynomial equations which can be automatically generated by MINI*SIM and CURVEFIT (a program to be described later). The basic calculation procedure is a

classical theoretical tray based tridiagonal matrix inversion procedure coupled with a modified Holland method to force material balance. The user must specify the distillate and reflux rate (plus the other requisite variables); MINI*TXT will predict the product distributions for the various components. During the course of the calculations the user may change most of the variables affecting fractionator performance-feed plate location, feed quality, reflux rate, distillate rate, etc. The number of theoretical trays in the column cannot be changed. Output from the program includes product distributions, vapor liquid and temperature profiles and condenser and reboiler duties. On option, the user may opt to also print the vapor and liquid composition profile and the K value profile.

MINI*TXT has been used to design many different columns including:

(1) "standard" depropanizers, debutanizers, etc.,
(2) low temperature demethanizers used in turbo expander plants,
(3) reboiled absorber including those used in the Ryan-Holmes process for high pressure CO_2 extraction, and
(4) crude oil stabilizers.

The program is not capable of handling the "standard" crude distillation unit with side stream strippers, pump arounds, etc.

The major limitation in MINI*TXT is the use of polynomials to predict the thermodynamic properties; these were used to expedite problem solution times not because of memory space limitation. For those systems, where the composition effects on the thermodynamic properties is small, this limitation should be of little concern. In other cases, the user may have to iterate through MINI*SIM several times to get a proper definition of the thermodynamic properties.

CURVEFIT

The CURVEFIT programs fits user supplied K value and enthalpy data to the appropriate form for use in MINI*TXT. Standard least squares fitting procedures are used. Each polynomial is tested for the presence of a maximum or minimum in the interval of fit and the user can either change the form of the fitted equation or change the data being fitted. Input to CURVEFIT is either by a MINI*SIM prepared data file or a user prepared data file. On option, CURVEFIT can produce an output file of equation constants which can be read by MINI*TXT.

TRAPP

This program was written by J. F. Eley and H. Hanley (5); it predicts the viscosity and thermal conductivity of multicomponent hydrocarbon systems over a wide composition - temperature - pressure range based on modified corresponding states procedures. The TRAPP program has been released through the GPA; we are simply making it available to our clients as a service.

TRAYSIZE

TRAYSIZE is a program which can be used to make a preliminary estimate of absorption or distillation column diameters. It is based on the Fair method (9).

SGP

SGP predicts the phase behaviour of acid gas/amine systems based on a modified Kent and Eisenberg model (10). MEA, DEA, DiPA and DGA can be handled by the program.

THE FUTURE

We are currently working on several new programs/systems for use on both the 8 bit processor, 64 K microcomputer and the new 16 bit processors. Among the programs are:

1. MINI*H2O - a program which will predict the phase behaviour of water-hydrocarbon-acid gas-methanol/glycol systems. This program will be based on the PFGC equation of state (4) and will be capable of predicting partial miscibility in all phases.

2. MINI*HEX - a program to perform preliminary sizing of shell and tube heat exchangers (2) and aerial coolers (15). Only single phase fluids can be considered at this time.

3. APM - a program to predict the performance of amine sweetening units (13). The amines handled by this program are MEA, DEA, DiPA and DGA. Acid gas-amine equilibrium is described by a modified Kent and Eisenberg approach.

4. MINI*VENT -a program designed to analyze high pressure relief systems which can involve two phase flow. A completely rigorous thermodynamic analysis is coupled with the Lockhart and Martinelli (11) two phase flow pressure drop procedure.

5. MINI*ADS -a program for the transient analysis of absorption equipment - mol sieve dehydration, silica gel hydrocarbon dew point control systems, etc.

6. PRO*CONT - a process control system analysis/design and operator training program. The transient analysis of processes is modeled using a series of quasi steady state calculations; equipment response to upsets are predicted.

7. MAXI*SIM -a steady state process simulation system which combines the interactive features of MINI*SIM with features of the traditional batch style simulator. Complete processes with recycle can be simulated in a completely interactive mode and a partially or completely batch style mode. The user can switch from one mode to another with simple mnemonic commands. The program will be based on the SRK method of predicting thermodynamic properties.

All of these programs are being designed to exploit the interactive capabilities of the microcomputer systems. These programs are now in various stages of development ranging from initial debugging to final field testing before release.

SUMMARY

Several of the currently available micro-computer based programs and typical applications of these programs have been described. Projections for future program availability have been made. We believe that the microcomputer technology coupled with our programs offers a tremendous economic advantage to the chemical process industry.

TABLE I

EQUILIBRIUM CALCULATION OPTIONS

Type of Calculation	Independent Variable	Dependent Variable
Bubble Point	P_1 to P_2, P	T
Bubble Point	T_1 to T_2, T	P
Dew Point	T_1 to T_2, P	P
Flash	T_1, P_1	L/F
Flash	T_1 to T_2, T, P_1	L/F
Flash	T_1 to T_2, T, P_1 to P_2 P	L/F
Flash	L/F, P_1 to P_2, P	T
Flash	L/F, T_1 to T_2, T	P
Flash	H, P_1 to P_2, P	T
Flash	H, T_1 to T_2, T	P
Flash	S, P_1 to P_2, P	T
Flash	S, T_1 to T_2, T	P

TABLE II

EDIT FUNCTIONS

Command	Function
TI	Enter a problem description
TT	Enter a new temperature
PP	Enter a new pressure
SL	Save the calculated liquid stream
SV	Save the calculated vapour stream
SF	Save the feed stream
MV	Move a stream
OP	Change the designated calculation option
CH	Change the total enthalpy of a stream
CS	Change the total entropy of a stream
CF	Change the molar fraction liquid
PR	Change the output style
ID	Change the component identification
RN	Perform calculations in the immediate mode
FD	Enter a new feed stream
AD	Add two streams together
LI	List a stream
NW	Request a new problem definition
SX	Save the calculated liquid mol fraction as a stream
SY	Save the calculated vapour mol fraction as a stream
MU	Multiply a stream by constant
TC	Adjust the critical temperature of one or more components
PC	Adjust the critical pressure of one or more components
C6	Modify the C_{6+} fraction parameters
FL	Save the current values of the major arrays for a restart
C2	Check for CO_2 freezeout
AQ	Add energy to a designated stream
VP	Perform a Reid Vapour Pressure (RVP) calculation
EF	Predict an EFV Curve for a given stream
NM	Enter a name for a stream
HT	Predict the gross heating value of a stream
ZS	List the vapour and liquid phase compressibility factors

TABLE III

SAMPLE PROBLEM DATA ADAPTED FROM REFERENCE (8)

Feed Composition

Component	Mol %
C_1	60.88
C_2	7.38
C_3	5.03
C_4's (nC_4)	2.78
C_5's (nC_5)	1.96
C_6's (nC_6)	1.84
C_{7+}	20.13
Total	100.00

Phase Behaviour Data

Pressure, bars	Mol% Vapor Extrapolated Bubble Point
374.0	0.0
344.7	8.83
310.3	34.87
275.8	35.75
206.8	44.57
137.9	55.34
76.5	62.99

C_{7+} Data

LV%	T, C
2.8	50
5.0	75
9.7	100
17.9	125
25.4	150
31.8	175
35.8	200
40.8	225
46.2	250
54.7	275
57.3	306
62.0	336
66.5	366
71.0	392
71.3	419

SPGR 0.8363

Mol Wt 191

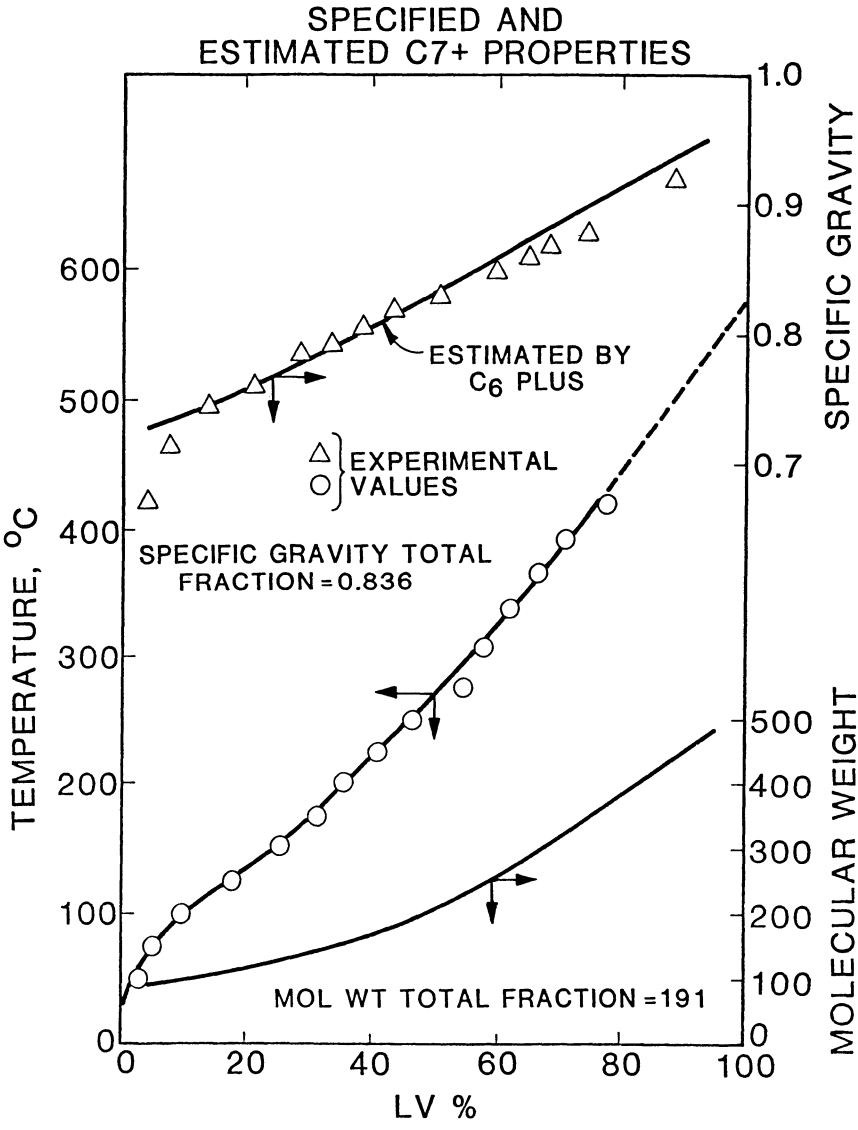


Figure 1

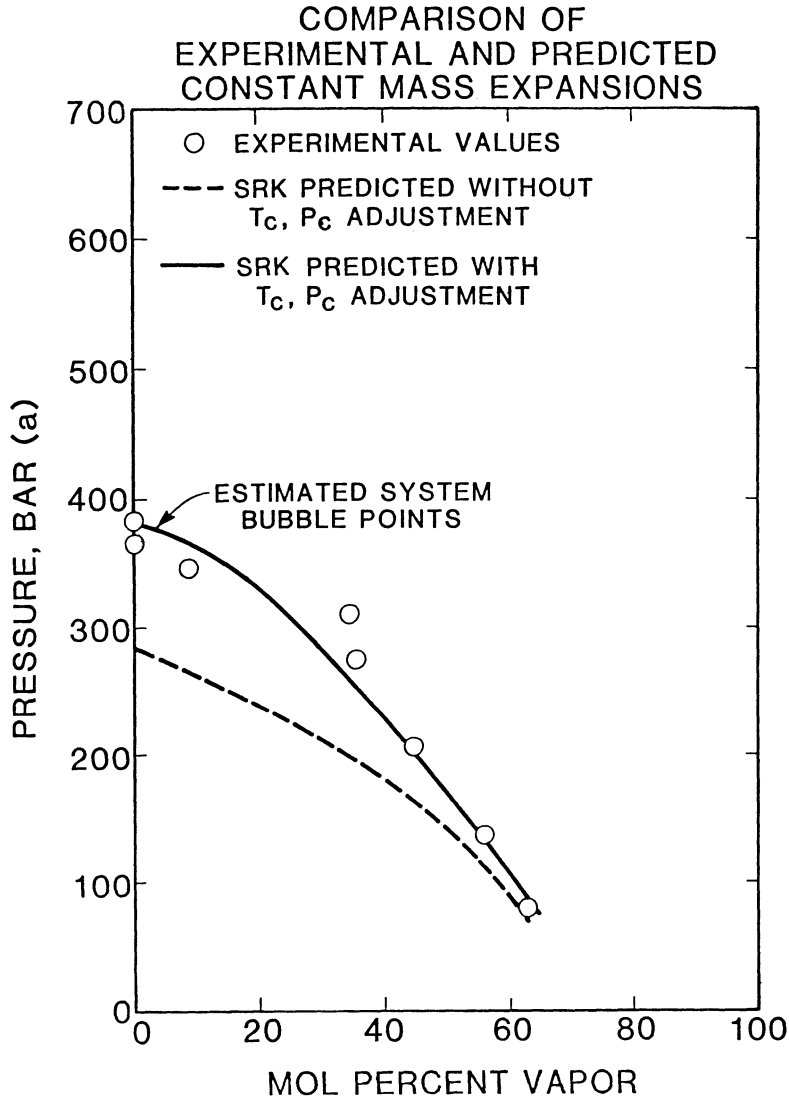


Figure 2

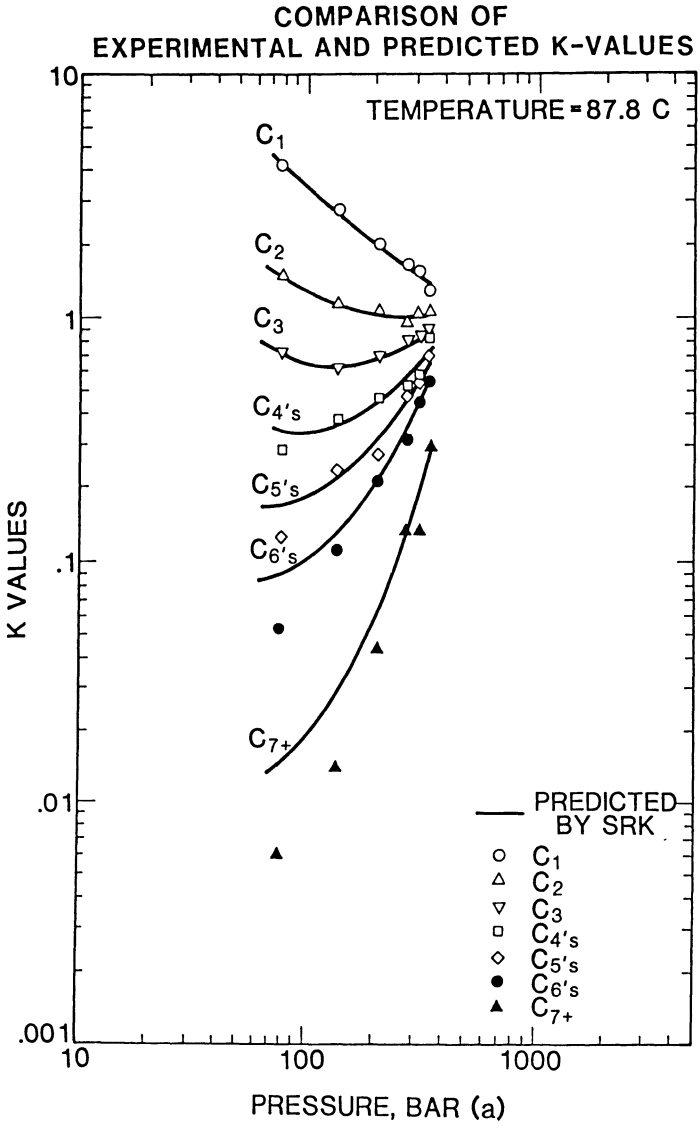


Figure 3

REFERENCES

Akashah, S., Ph.D. dissertation, School of Chemical Engineering, Oklahoma State University, Stillwater, Oklahoma (1980).

Bell, K. J., "Process Heat Transfer", Class Notes, School of Chemical Engineering, Oklahoma State University, Stillwater, Oklahoma (1981).

Charonsumbut, A., C_6+ Characterization, MS Report, School of Chemical Engineering, Oklahoma State University, Stillwater, Oklahoma (1978).

Cunningham, J. R. and G. M. Wilson, Proceedings of 54th Annual Meeting GPA, Gas Processors Association, Tulsa, Oklahoma (1974).

Ely, J. F. and H. J. M. Hanley, "Transport Properties Prediction Program Computer Program TRAPP User Guide", Thermophysical Properties Division National Engineering Laboratory, National Bureau of Standards, Boulder, Colorado (1982).

Erbar, John H., GPA Research Report No. 13, GPA, Tulsa, Oklahoma (1977).

Erbar, John H., "MINI*SIM User Documentation", Chemical Process Consultants, Inc., Stillwater, Oklahoma (1981).

Evans, R. B. and D. Harris, Chemical Engineering Data Series, Vol. 1, No. 1, p. 45 (1956).

Fair, J. R., Chapter 11 in Smith, B. D., "Equilibrium Stagewise Processes", McGraw-Hill, New York, New York (1967).

Kent, R. L. and B. Eisenberg, Hydrocarbon Processor, No. 2, p. 87 (1976).

Lockhart, R. W., and R. C. Martinelli, Chemical Engineering Progress, Vol. 45, No. 39 (1949).

Soave, G., Chemical Engineering Science, Vol. 27, No. 6, p. 1197 (1972).

Vaz, R., Ph.D. dissertation, School of Chemical Engineering, Oklahoma State University (1980).

Wilson, A., R. N. Maddox, and J. H. Erbar, Oil and Gas Journal, July (1978).

_____, GPSA Data GPA, Tulsa, Oklahoma (1972).

SOFTWARE ASPECTS OF A MULTI-MICROPROCESSOR SIMULATION SYSTEM

Peter L. Harrod and Derek C. Barker

Department of Electrical Engineering and Electronics,
University of Manchester Institute of Science and Technology.

INTRODUCTION

Objectives

During the last decade the cost of energy has become an increasingly important factor in building design, and increasing resources are being allocated to the measurement, analysis and prediction of the thermal characteristics of buildings. Presently available methods and facilities are adequate for the mainly research oriented measurement and analysis purposes. Prediction, however, poses very different problems because its main objective is to enable the builder/architect to balance the extra costs of energy-saving designs against the actual savings in energy consumption. Hence the requirement is for an essentially interactive computing system on which he can rapidly assess the effects of design modifications, in terms of the annual fuel bills and the cost of materials.

This type of performance is at present only available through single-user use of very large, expensive to buy and to operate, mainframe computers such as the CDC 7600 or the CRAY-1, or alternatively through innovative designs such as UMIAC-80 [Barker, 1979; Barker and Harrod, 1980; Harrod, 1982], a Distributed Array of Microprocessors designed and built in the Department of Electrical Engineering and Electronics, UMIST, for the Building Research Establishment, Department of the Environment. It is the purpose of this paper to discuss the software aspects of the design of UMIAC-80.

Statement of the problem

(i) Organisation We consider a building as consisting of elements : walls and floor/ceilings. Each element in turn consists of sections (brick, plaster, air gap, etc.) according to its construction (Figure 1).

The problem then is the determination of the flow of heat through the elements when subject to air temperatures T_a, T_b

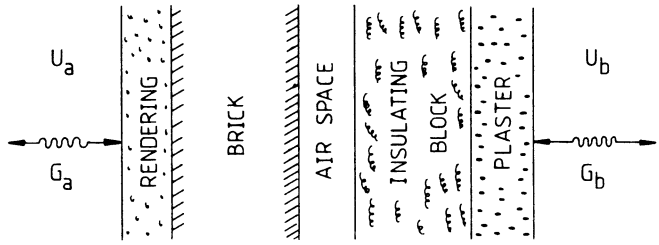


Figure 1 Typical element (house external wall)

and radiation gains G_a, G_b at each element surface. The conduction process is governed by the diffusion equation

$$T_{xx} - (c\gamma/\lambda)T_t = 0 \qquad (1)$$

where T, x and t are respectively absolute temperature, distance and time, c is heat capacity, γ is density and λ is the coefficient of thermal conductivity.

The numerical procedure used for solving Equation 1 is not of importance in the design of UMIAC-80 (or indeed of most other computers), though to clarify the problems of data flow and of finite word length, and to test system operation, a conventional finite difference approach was adopted.

(ii) Scale of the problem Whichever method of solution we adopt we are in principle concerned only with the solution of a set of linear simultaneous equations at each element. However, the size and complexity of a real building introduce many difficulties. An average house may well have 50 or more elements, and as for each element we may have 15 or more simultaneous equations (i.e. 15 or more nodes), we have to solve upwards of 750 simultaneous equations. A fairly minor office block may well have ten times this number.

It is seldom practicable to consider the direct solution of the full set of equations for any but the smallest of buildings. Thus a 1000 element office block may require the simultaneous solution of 15000 or more equations for each of the 17520 time steps (at half-hour intervals) throughout the year. Assuming efficient storage of the non-zero elements of the (tri-diagonal) matrix the required memory is of the order of $1500^{3/2} = 1{\cdot}8 \times 10^6$ words. Of course, as the equations increase in complexity with the introduction of windows, doors, radiators, etc. efficient storage becomes less possible and the required fast access memory tends to $15000^2 = 225 \times 10^6$ words. Thus a better

approach is to obtain the solution for each wall (or each room) on an individual basis, working either sequentially or simultaneously through the building. The required working memory is greatly reduced but it becomes necessary to use an overall iterative approach to establish the overall energy balance.

(iii) Parallel processing The choice between a sequential or a simultaneous solution of the heat flows in the elements of a building determines the type of computer to be used. A parallel approach to the essentially parallel problem is attractive in that the overall processing time is to a large extent independent of problem size, but until recently only the sequential, von Neumann architecture, computer was available.

The microprocessor now enables us to seriously consider parallel processing. It is cheap and systems can be implemented with progressively lower chip counts (at present about 30 chips per processor). Furthermore the same technology permits us to use cheap fast-access memory in large quantities. Thus a relatively cheap 100 microprocessor model could at present have 1·6 Mega words of fast access memory, and 6·4 Mega words is becoming feasible.

THE STRUCTURE OF UMIAC-80

For the design of UMIAC-80 the authors adopted a simultaneous processing of individual elements method, allocating a processor to each wall or floor/ceiling of a single floor level of a building. This is not full parallelism on two counts : each processor is solving a number of equations on a sequential basis; and only one floor level is dealt with at a time. These constraints were adopted to keep the size of the array within practical limits, and in practice only the second will increase processing time. This is because each processor may use fast direct methods (e.g. Gaussian Elimination) to solve the limited number of simultaneous equations corresponding to an element, whereas in a fully parallel system the slow Jacobi iterative scheme must be implemented throughout the whole building. A further constraint was the decision to limit the resolution of the mapping of processors on to a building level to that achievable with a fixed, limited, number of processors.

The thermal radiation exchange between surfaces is governed by their temperatures. Hence each element processor must receive the surface temperatures of all elements with which it shares a room, and it must in turn transmit to them its own surface temperature. Hence at each time-step in the computation each processor must transmit at least one, and generally two, surface temperatures. In order to accelerate this data exchange process the authors have adopted a local data bus network arranged so as to maximise the parallelism of data transfer .

To avoid unnecessary complexities in this local bus all

processors have been allocated fixed positions in the array. The basic model of one level of a building then consists of modules as shown in Figure 2. Thus all local bus data paths

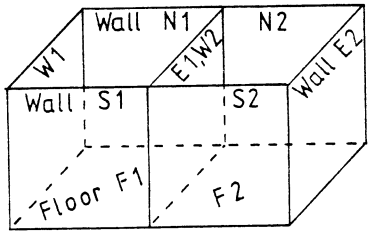


Figure 2 Building level modules

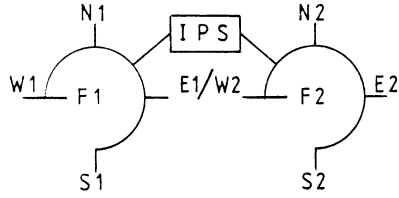


Figure 3 Local bus structure

are predetermined, interconnecting five processors as shown in Figure 3. We have two types of element processors, one for floor/ceilings and one for walls.

During the simulation process the operator will have to determine each room representation, indicating those elements which are to be used. The unused elements are deemed transparent, allowing the flow of radiation data to pass through unimpeded. Thus in the multi-module room of Figure 4 complex data paths are formed.

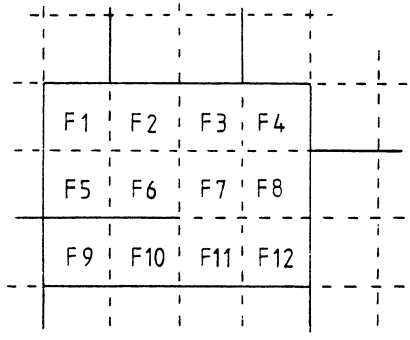


Figure 4 Multi-module room

UMIAC-80 HARDWARE

The system hardware can be divided into two parts : the host computer and the array of Processing Elements (PEs). The host

computer is a conventional microprocessor development system which has been enhanced by the provision of an Input/Output (I/O) board that forms the interface to the array. All program code and data is transferred between host and array over a common bus, which is driven by I/O ports under program control. Daisy-chaining of the handshake control signals allows global transfers to occur.

Each PE in the array is a self-contained microcomputer with its own 4 MHz clock, private volatile and non-volatile memory, and I/O, and it can thus execute programs autonomously. A block diagram of the PE hardware is shown in Figure 5. The Z80A

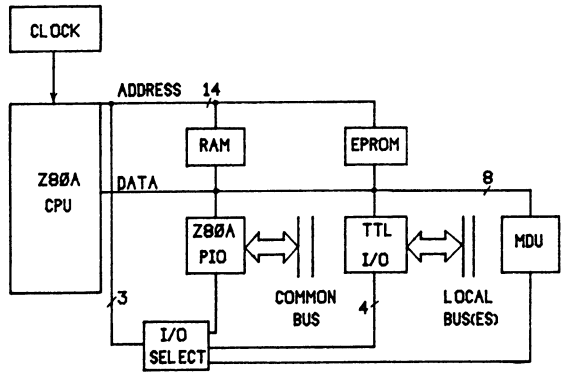


Figure 5 Processor hardware and bus structure

Central Processing Unit (CPU), a third generation 8-bit microprocessor, offered the best performance/cost ratio at the initial design stages (1978), and with 1 K Byte of EPROM, 16 K Bytes of Dynamic RAM and a 16-bit Multiply/Divide Unit (MDU) it forms a PE of reasonable performance in arithmetic-intensive processing. Input/Output to both the common and the local buses is under program control, using a fully interlocked handshake protocol. Data transfers thus depend upon co-operation between the transmitting and the receiving PEs, a fact that must be considered when developing the programs for each PE.

SOFTWARE REQUIREMENTS

It cannot be assumed that the potential users of UMIAC-80 will have an in-depth knowledge of computer operation, so the interface to the user has to effectively isolate him from the intricacies of the system hardware. However, applications programmers need to have a knowledge of the system's parallel processing structure in order to be able to use this special purpose hardware effectively. At whatever the level of programming or use of the machine, a comprehensive suite of software is required such that applications programs may easily be developed. The

intention is to provide a friendly software environment in which the user is encouraged to add or modify programs to suit his or her own particular requirements, without being concerned with the details of file handling, I/O device drivers, etc. These details are contained in a set of system library routines.

The host microcomputer, a Zilog MCZ 1/05 development system, is a cost-effective solution to providing system control and user interface in one unit, while at the same time providing an environment for developing Z80 software. To ensure software compatibility and efficiency of execution on the Zilog hardware, only Zilog software products are used. These range from the Z80 assembly language, used where minimisation of execution time and/or memory utilisation is important or when specific Z80 features need to be accessed (e.g. the interrupt structure), to PLZ/SYS, Zilog's high level language, used for most of the system and applications software. Interconnection of program modules written in different members of the PLZ language family can be combined at link-time [Bass, 1978] to produce an executable procedure file that achieves the desired balance between execution time and code size.

The software developed for UMIAC-80 can be divided into two distinct but related parts : that required for the host microcomputer and that required for the processing elements in the array.

Processing element software

Bootstrap and monitor A bootstrap and monitor program (BOOT-MON) resides in the 1 K Byte EPROM (non-volatile memory). BOOT-MON implements standard monitor functions, such as loading and reading memory contents and starting or stopping program execution, in addition to bootstrapping, self-diagnostic and debug routines. This program also uses a small amount of volatile memory (RAM) to implement a stack and to store the processor's state vector. The remainder of the RAM is available for applications programs. The operation of BOOT-MON is outlined in Figure 6. (See over page)

On system reset all parameters and I/O devices are initialised and a self-test of the hardware is carried out. The result of this test is indicated by an LED on the edge of the PE's circuit board, thus allowing a visual inspection of the operational status of the system. The PE then remains in a HALT state, until a command interrupt from the host is received. An application program may similarly be interrupted to service a command from the host, and thus the processor state vector must be saved, and then restored when the command has been serviced. This feature allows the program activity in, or the memory and register contents of, a particular PE to be examined without upsetting operation of the parallel processor as a whole.

The monitor is provided with facilities to implement

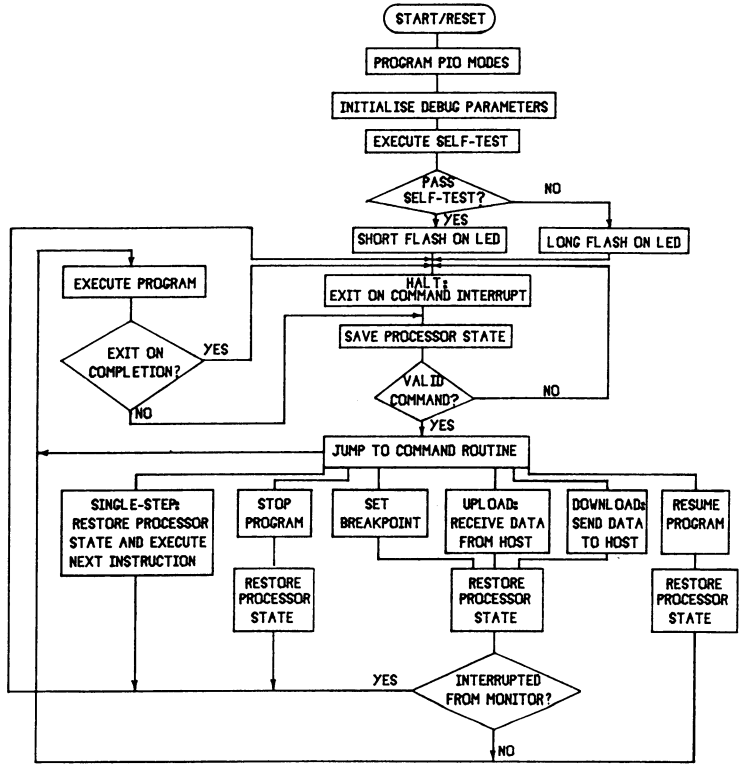


Figure 6 BOOTMON

the important trace and debug functions on applications programs. These facilities include stopping a program at a breakpoint, displaying and modifying the contents of CPU registers and of memory, single-stepping the program and resuming program execution (after a breakpoint has been reached or the host has caused a command interrupt). A number of unique trace and debug considerations arise on a multiprocessor system : for instance the PEs may each be executing an individual program, so it is essential that tracing a program flow on one PE should not affect the operation of any other PE even though their operation may be interrelated due to co-operation in data transfers.

UMIAC-80 has been designed to facilitate the debugging of software on the multiprocessor : commands may be sent globally or individually to the PEs so that, for example, their breakpoints may be set, or they may be single-stepped, together or individually. Furthermore, the daisy-chaining of the common bus control signals allows any one PE to signal a particular condition. For instance the HALT Acknowledge lines for each PE are

daisy-chained such that any PE, by halting on encountering an error condition, can cause all other PEs to stop at that point, allowing the host to examine the complete machine state which existed when the error occurred.

Applications programs Programs that are to run on the PEs must conform to certain specifications, which are checked by the host before downloading occurs. The only preparation required of the monitor is to initialise the stack pointer before control is passed to the applications program. Programs may be written in any language that can be translated into Z-80 machine code and linked into executable procedure files. Although it would be possible to run interpreted code this has so far been rejected on the grounds of reduced efficiency.

Two major library modules have been provided to assist the applications programmer : one is a suite of interprocessor data transfer control subroutines that ensures secure communications on the local buses (a fully interlocked handshake protocol is used), and the other is a suite of floating point arithmetic functions which make use of the hardware multiply/divide unit provided on each PE.

Host computer software
The function of the host computer is to control and co-ordinate the various components of the UMIAC-80 system while carrying out a simulation. To do this it must provide the man-machine interface, control the array of PEs and control peripheral devices (line printer, display panel, etc.) in addition to providing software development facilities and mass storage of data.

To enable the commissioning and testing of the system to be performed effectively the initial emphasis was on developing software for the control of the array of PEs, and on developing the user interface required to make use of this software. The program U80COMM and its associated modules constitute the array control software, and allow communication between the host and one or more PEs over the common bus. The development of U80COMM and BOOT-MON proceeded in parallel due to their close interaction; U80COMM exploits the features of BOOT-MON to provide the operator with a comprehensive screen-based program execution and debugging environment for the control of the array of PEs. The modular structure of U80COMM allows other programs to make use of its functions, such that, for instance, a simulation run can consist of a sequence of load, execute and read commands. More recently application programs have been developed, as described later.

Problem specification For each new problem the geometry and construction of the building has to be specified by the operator. In the context of UMIAC-80 this entails selecting which floor and wall elements are to be used to model the building,

and specifying the dimensions and material composition of each of these elements. A structured data base, consisting of data files, has been designed [Okoyo, 1982] to store this information so that a building need be specified once only. Element selection is facilitated by making use of the line graphics available on the visual display unit (VDU). A suite of VDU control routines were written in PLZ/SYS to enable the features of the VDU to be accessed by means of simple system calls. Information relating to the weather conditions over an annual cycle is a further requirement of the model; it is stored on disk and appropriate data is extracted and transferred to external wall elements as required. Provision of software to accomplish the above functions has been a major undertaking. This software requires further development, with particular reference to the flexibility of the user interface.

Output data Data required by the user would typically be the building's local and total energy consumption profiles and the room temperature profiles. All this data must be collected from the array of PEs by the host, be processed as necessary, and then be presented in the form required on the VDU or printer, or else be stored on disk. User requirements will naturally vary, and consequently all the above functions are provided as a library of system routines.

Algorithm implementation The solution of the heat flow equations required for modelling the thermal performance of a building can be performed by a variety of techniques. An implicit finite difference scheme, using a Gaussian Elimination algorithm for the heat flow within an element and a Jacobi iterative scheme for the overall solution at each time step, has been implemented on UMIAC-80 to enable system testing. However this at present provides only a very simple model of the building's thermal performance, and it is intended that the user will develop more advanced algorithms for use on the system. Existing algorithms written in a high level language may be adapted by partitioning of the software and incorporating the necessary control and data transfer routines. This has been successfully achieved for a program that uses a new method of solution based on normal co-ordinates [Gough, 1982], as discussed later.

System operation The simulation process on UMIAC-80 is characterised by two distinct phases of operation : computation and communication. This has allowed a simple master-slave control structure to be implemented, by means of which the host has control over every aspect of system operation. The host issues commands over the common bus to the PEs in the array to achieve transfer of data and code, execution of programs, etc. A typical sequence of events would be as follows :

1. Specify building geometry and construction.
2. Check completeness of data (repeat 1 if necessary).

3. Load appropriate PEs with solution algorithm and relevant data.
4. Command PEs to execute algorithm for N time-steps.
   PEs perform local data exchanges under local control; host controls data exchanges (if any) between non-neighbouring PEs.
5. Host reads results for storage or display.
6. Optional operator interaction .
7. Repeat from 4 until end of simulation period.

The host makes use of the global control signals and daisy-chained handshake lines to control the array of PEs. For example, one daisy-chained line indicates that convergence (of the overall Jacobi iteration) has occured in all rooms of the building, whereas another indicates that the computation phase has been completed on all PEs, so the host can load results.

PARALLEL PROCESSING SOFTWARE

The special purpose hardware structure of UMIAC-80 imposes a number of constraints upon the software that can be developed for the system. In particular, separate programs are required for each of the three types of processor (host, floor and wall). These programs can execute concurrently and therefore some method is required for their control and synchronisation. At present a manual method of program decomposition is used, such that programs which must include explicit control and synchronisation instructions are written for each processor type. This is a straightforward process, as will be illustrated by means of an example.

Program development for a single sequential processor can be thought of as a one-dimensional process, one instruction following another along a vertical (time) axis on the page. Development of a parallel program for a set of concurrently executing processors is, in contrast, a two-dimensional (time-space) process. Instructions which must be executed sequentially are written along the vertical (time) axis while those that can be executed concurrently are written along the horizontal (space) axis, as shown in Figure 7. (See over page)

This procedure has been adopted successfully for the decomposition of a program which models the performance of a single storey nine-roomed building using a new efficient numerical method of solving Equation 1, based upon normal co-ordinates [Gough, 1982], and which was originally written for a single processor. In the original version the calculations involved in determining heat flows through the 24 walls of the building and in obtaining a heat balance in each of the nine rooms were achieved by a series of repetitive 'DO' loops. On UMIAC-80 each 'DO' loop can be distributed as appropriate over the 24 wall PEs and nine floor PEs of the array, thereby reducing the computation time proportionately.

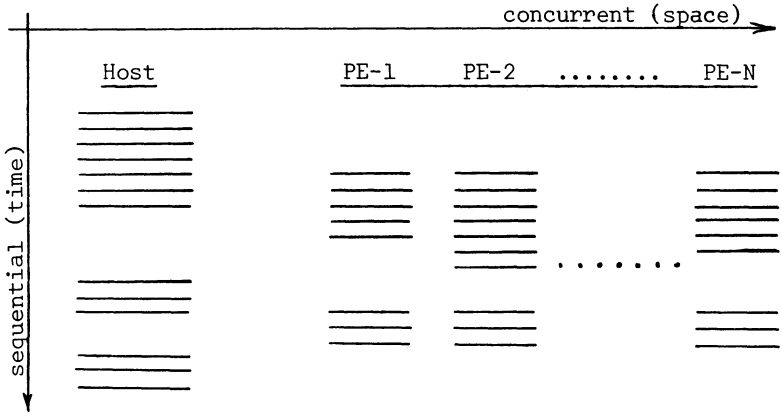


Figure 7 Concurrent (two-dimensional) programming

The first stage is to identify the correspondence between data declarations in the single processor program and processors in UMIAC-80, and to copy these into the appropriate columns on the page. The program code can then be partitioned in a similar fashion. Data transfers between floor and wall processors will replace certain assignment statements in the original version. Finally extra statements are added to the host program to provide control of the array of PEs. A parallel program implementation is outlined in Figure 8.

This manual method of program partitioning has been effective in the development stage of UMIAC-80, when the programmer has been expected to have some knowledge of the hardware structure of the system and the constraints which it imposes. Control of the PEs is achieved by command interrupt from the host, and synchronisation between PEs is achieved through the exchange of data. However, a more efficient method of programming the system is now desirable.

One approach is to make use of a "parallel language" which incorporates special features and a notation that facilitates the development of concurrent programs. Pascal-Plus [Welsh and Bustard, 1979] is one such language, which extends the familiar Pascal language with the process, monitor and condition constructs that are required for programming a system of parallel activities. The use of such a language ensures that control and synchronisation are rigorously described, making it easier to construct and verify the correctness of programs which consist of independent activities that interact in time.

The disadvantage of this approach is that the programmer is still required to write a parallel program and may in so doing be required to learn new programming languages and methods. To persuade the programmer to take advantage of the hardware parallelism available the complexities of the parallel

ORIGINAL SINGLE PROCESSOR PROGRAM	HOST	FLOOR NO.1 ....FLOOR NO.9	WALL NO.1 .... WALL NO.24
DATA DECLARATIONS			
Stepsaday, stepno BYTE Q_{max}, T_{stat} REAL T_{new}, T_{out} REAL itcount INTEGER	Stepsaday, stepno BYTE	Q_{max}, T_{stat} REAL itcount INTEGER	T_{new}, T_{out} REAL
PROGRAM CODE			
initialise	initialise send-floor-code send-wall-code send-floor-data send-wall-data send-execute-command		
DO J := 1 DO update (wall_state [J]) IF J = 24 THEN EXIT FI J += 1 OD K := 1 DO compute (room_temp. [K]) converge-test IF K = 9 THEN EXIT FI K += 1 OD IF converge = TRUE THEN EXIT FI OD	DO !host control! OD	DO !receive updated parameters from walls! !calculate new room temp.! !send room temp. and radiation parameters to walls! !Test for convergence! IF all-converge = TRUE THEN EXIT FI OD	DO update - wall state !send updated heat flow and radiation parameters to floor! !receive room temp. and radiation parameters from floor! IF all-converge = TRUE THEN EXIT FI OD

Figure 8 Outline of Parallel Program Implementation on UMIAC-80

programming method should be minimised. There is clearly a requirement for software tools that enable the parallelism in a program to be extracted automatically.

A 'parallel program development system' has recently been described [Grasso et al, 1982] that achieves this objective, albeit for a multiprocessor system that is to be used in a dedicated real time application. This system enables a parallel program suitable for a multiprocessor to be produced automatically from a sequential program written for a uniprocessor. The program is first divided into a number of tasks; the order of execution of these tasks and those which can be executed in parallel are then determined by identifying data dependencies between tasks. This information is then used by the operating system to exploit the potential parallelism in a program. It is intended that the next stage of development of UMIAC-80 will include a facility of this kind.

CONCLUSIONS

UMIAC-80 has proved to be a cost-effective solution to speeding up the computations involved in thermal modelling problems. Special purpose software has been required to take advantage of the hardware parallelism available. Extra facilities have been provided on the host microcomputer to enable the array of processing elements to be controlled effectively. These facilities, in combination with a monitor program on each processing element, provide a useful multiprocessor software development environment. At present a simple partitioning of a sequential program allows parallel programs to be developed. Parallel languages and methods that extract the parallelism in a program automatically are developments that will become of increasing importance in the future.

ACKNOWLEDGEMENT

This work was carried out for the Building Research Establishment, Department of the Environment. The authors are grateful for their support and encouragement. The views expressed are those of the authors.

REFERENCES

Barker, D.C. (1979) A Distributed Array Processor for the Simulation of Heat Transfer in Buildings. Numerical Methods in Thermal Problems. Pineridge Press, Swansea, pp.943-953.

Barker, D.C. and Harrod, P.L. (1980) A Distributed Array of Microprocessors for Evaluating the Thermal Performance of Buildings. Proc. CAD 80, IPC Science and Technology Press, Guildford, pp.615-626.

Bass, C. (1978) PLZ : A Family of System Programming Languages for Microprocessors. Computer, 11, No.3.

Gough, M.C.B. (1982) Modelling Heat Flow in Buildings : an Eigenfunction Approach. Ph.D. Thesis, Cambridge University.

Grasso, P.A., Forward, K.E. and Dillon, T.S. (1982) Operating System for a Dedicated Common Memory Multimicroprocessor System. Computers and Digital Techniques, Proc. IEE, 129-E, pp.200-206.

Harrod, P.L. (1982) An Array of Microcomputers for Modelling the Thermal Performance of Buildings. Ph.D. Thesis, UMIST.

Okoyo, H.O. (1982) A Program for Modelling Building Structures. M.Sc. Dissertation, UMIST.

Welsh, J. and Bustard, D.W. (1979) Pascal-Plus - Another Language for Modular Multiprogramming. Software - Practice and Experience, 9, pp.947-957.

THE APPLICATION OF GOAL PROGRAMMING FOR DETERMINATION OF OPTIMAL CONCRETE MIXES

Nenad Mladineo, Boris Ramljak

Gradjevinski Institut, Faculty of Civil Engineering
University of Split, 58000 Split, Yugoslavia

INTRODUCTION

Most optimization problems can be mathematically formulated as the problem of bound extremes with linear or non-linear relations. With linear relations linear programming (LP) models can be used and these have been successfully used to solve a great number of practical problems. Determination of the optimal mix for obtaining a finished product with the required properties and quality is one of those problems by what is known in literature as the mix model. The model is based on the fact that in some mixes the quantity of one or two ingredients can vary independently within the given constraints assuring that the quality of the finished product will satisfy the given requirements. Using these constraints it is possible to determine such ratios between ingredients to ensure optimal cost efficiency or optimal economic or technical-technological solutions to produce the final product (Bonacci O., Mladineo N., 1981).
In a great number of products (for example concrete mixes), a change can be made in the quantity of one ingredient only if we change the proportions of all other ingredients, and the permitted tolerances cannot be used as room (optimization area) for optimization. Such problems exclude the application of the standard mix model, because there is no standard optimization area within which our solution is sought. Some experiments, pracitacl as well as theoretical, carried out at the Faculty of Civil Engineering in Split, have pointed to the possibility for successful solution of such problems concerning mixes by using goal linear programming (GP). (Mladineo N., Krstulović P., Ramljak B., 1982). This method makes it possible to find optimal ratios between the ingredients in a concrete mix satisfying the given "ideal or reference curve" obtained by experiments. The essence of the method lies in the fact that the optimization area for a single ingredient has been replaced by a "reference set of data" which is presented in the model as the goal to be

obtained given the possible technological tolerances. Thus in the new mathematical model, instead of constraints presented by ineqations (inequalities), equations are formulated each of which represents one goal which can be obtained with minimum deviations. As goal programming is one of the methods of multi--criteria optimization, it is possible even in the developed model to look for a complex optimum for a concrete mix according to two criteria: economical and technical-technological.
As will be explained in the next chapter the sensitivity of the qualitative properties of concrete to changes in the technological composition calls for special accuracy in defining the technological relations in the mathematical mix model.

TECHNOLOGICAL REQUIREMENTS

The tender document and the design of concrete structure are used to define the quality requirements to be satisfied by fresh or hardened concrete. The composition of a concrete mix which meets these requirements is generally established by experiment. The ideal or reference composition has to be uniquely defined for each class of concrete and for each sort of basic material. The basic parameter employed to define the quality of fresh concrete is its workability, i.e. its capacity to occupy all the space in the mould. As the workability parameter we most frequently use data on the concrete mix consistency obtained by one of the standard methods, for example slump according to the Abrams or Vebe methods.
One constant alone,eig. slump (S) or Vebe (VB), does not define the workability of a concrete mix. Workability can be successfully defined using a multiple parameter rheological model or a pair of S and VB data. The composition of a concrete mix is defined by a volume model. Mixes with different volume models can all have the same S, but each of them has a different VB. Consequently, only one volume model of the mix containing the avaliable basic ingredients coresponds to the given pair S,VB. Figure 1 presents the volume model of one mix in a form suitable for further treatment. This Figure actually represents the "granulometric composition" of a complete concrete mix.
Fresh concrete consists of two basic ingredients:

- paste
- aggregates

The quality and composition of the paste are presented on the ordinate with the apscissa being 0,125 mm. On the ordinates for the other sieve openings we present the volume ratios between the concrete mix ingredients passing through the given sieve for one unit of the mix volume. The compressive strength of hardened concrete can be defined accurately enough by a water-cement ratio. This water-cement ratio is determined experimentally for each class of concrete for a given S. Mixes of different classes contain different quantitites of cement, i.e. a different amount of paste. Consequently, they have different volume models and corresponding values VB. Differences in rheological characteristics defined by a S and VB pair can be

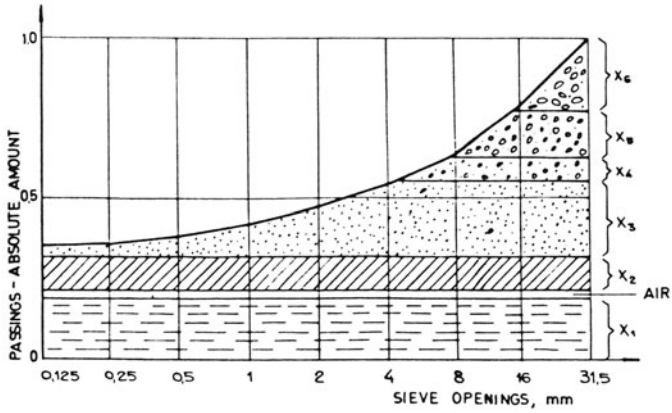


Figure 1 - Granulometric composition of concrete

significantly reduced if the proportion of paste is either too high or too low with regard to the granulometric curve it can be compensated by changing the quantity of the aggregate which is most similar to cement in its individual granulometric composition.
The technological requirements presented can be expressed by a system of equations:

$$a_{11}x_1+a_{12}x_2+a_{13}x_3+a_{14}x_4+a_{15}x_5+a_{16}x_6 = b_1 \quad (1)$$

$$a_{21}x_1+a_{22}x_2+a_{23}x_3+a_{24}x_4+a_{25}x_5+a_{26}x_6 = b_2 \quad (2)$$

$$a_{31}x_1+a_{32}x_2+a_{33}x_3+a_{34}x_4+a_{35}x_5+a_{36}x_6 = b_3 \quad (3)$$

$$a_{41}x_1+a_{42}x_2+a_{43}x_3+a_{44}x_4+a_{45}x_5+a_{46}x_6 = b_4 \quad (4)$$

$$a_{51}x_1+a_{52}x_2 = 0 \quad (5)$$

$$a_{62}x_2+a_{67}x_7 = 0 \quad (6)$$

where:
x - proportions of ingredients in one unit of the concrete volume
x_1 - water
x_2 - cement
x_3, x_4, x_5, x_6 - aggregate
x_7 - additive
Equation (1) shows the requirement that the grainsize coefficient of the reference grading curve of concrete for sieves ranging from 0,125 mm to sieves with a maximum grainsize of (d_n)

of the aggregate used to compensate for the quantity of paste, should be equal to the sum of the parts represented by individual ingredients:

$$a'_{11}x_1+a'_{12}x_2+a'_{13}x'_3+a_{14}x_4+a_{15}x_5+a_{16}x_6 = b'_1 \qquad (1')$$

The value of x_3 should be lower than x'_3 and the following correction is introduced:

$$x'_3 = \frac{1-c}{a_{1d}\,c}(x_1+x_2) + \frac{x_3}{c} - \frac{1-c}{a_{1d}\,c}(P_{rn} - V_z) \qquad (1'')$$

where:

a'_{1k} and a_{1k} - grainsize coefficient of individual ingredients, in sieves with openings ranging from 0,125 to d_n

b'_1 - grainsize coefficient of the reference grading curve of concrete, in sieves with openings from 0,125 to d_n

P_{rn} - passings through sieve d_n on the concrete reference grading curve

V_z - absolute air content

a_{1d} - passing of substitution fraction through sieve d_n

After the correction, we have the equation (1).

Coefficient c was determined by the authors according to their theoretical and experimental investigations. Characteristic values for the system of sieves in Figure 1 are:

d_n	c
4 mm	0,68
1 mm	0,41
0,25 mm	0,097

Coefficient c depends only on the sieve system and on d_n. Equations (2), (3) and (4) express the condition that the passing on the reference grading curve of concrete (b_2,b_3,b_4) through sieves with 8,16 and 31,5 mm openings, reduced for the content of air, should be equal to the sum of the parts represented by individual ingredients. Coefficients a_{ik}, i=2...4, k=1...6, are passings through the sieves of individual ingredients. Equation (5) represents the condition (requirement) that the water-cement ratio of a certain concrete class should be constant in relation to its volume. The additive is dosed in % by cement mass, as defined by equation (6) in terms of volume relations.

CONCRETE PRODUCTION

Concrete production includes all those activities carried out during the production of the finished structure, including the production of concrete and the removal of the shuttering and scaffolding, but not the cost of the shuttering and scaffolding.

The basic output cost (C_{po}) per volume unit of placed concrete is established by financial calculations for a basic volume model of concrete with defined rheological properties; for example S=6 cm, VB=3 sec (reference grading curve of concrete) with the basic water content, x_{10}=0,188.

When defining the post of production as the function of parameters S and VB, we started from Lyse´s rule that parameter S is linearly dependent on the uknown x_1 (Powers T., 1968). This rule is acceptable if a volume contribution of grains smaller than 0,125 mm is found within limite ranging from 0,09 to 0,17. Consequently, the cost of concrete production can be defined as the function of the unknown x_1 in a system of equations (1) to (5). According to investigations to date, the cost of concrete production increases in a geometric progression as the amount of water in the mix decreases. The expression for the cost of production is:

$$C_p = C_{po} \left(1 + a_1 \frac{q^n - 1}{q - 1} \right) \qquad (7''')$$

where:

a_1 - coefficient for increases in the cost of placing, if the value of the unknown x_1 increases from x_{10} to x_{10}+0,01

q - quotient of cost increase (increment) for an interval of the unknown x_1 being 0,01

n - number of intervals, where for x_{10}...n= 0

Values a_1 and q depend on the type of structure and should be determined by analysing the organization of concrete production experimentally. Value a_1 ranges from 0,05 to 0,15 and the value of q from 2 to 4. Further computation can be simplified by introducing the unknown

$$x_8 = \frac{C_p}{C_{po}} = 1 + a_1 \frac{q^n - 1}{q - 1} \qquad (7'')$$

The unknown x_8 can generally be expressed as an exponential function:

$$x_8 = 1 + \frac{a_1}{q - 1} \left(q^{100(x_{10} - x_1)} - 1 \right) \qquad (7')$$

Strok gives a similar expression for the cost of energy when placing concrete by vibrations.

Function (7') is linearized by replacing it with a polygon which has apexes in the extreme points of an arbitrarily chosen interval of the unknown x_1. For the chosen values x_1 of two adjacent polygon apexes, the values x_8 can be computed using equation (7') and we can determine the coefficients of the equation for the corresponding line, which generally has the following form:

$$a_{71}x_1 + a_{78}x_8 = b_8 \qquad (7)$$

If we use a water-reducing agent we should determine experimentally the content of air, the basic quantity of wtaer x_{10} and the water-cement ratio for each class of concrete. We introduce the given coefficients in relation to those parameters into equations (1) to (7). If any type of plastificator is used we define experimentally only the basic quantity of water x_{10} and we introduce the given coefficients into equations (1) through (7).

The previously presented seven equations can be defined as goals which have to be achieved in order to satisfy the technological prerequisites for obtaining high quality concrete. Apart from the technological criteria present in determining the composition of the mix, we should consider the economic criteria, for each component (x_1 to x_8) has its cost:

c_1 - cost of water

c_2 - cost of cement

c_3, c_4, c_5, c_6 - cost of aggregates

c_7 - cost of additives

c_8 - cost of placing

If we define the costs we can establish the condition for obtaining the economic optimum, i.e. the goal equation has the following form:

$$c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6+c_7x_7+c_8x_8 = F(x)_{min} \qquad (8)$$

Analysis of the given goal function reveals the fact that the process of concrete production, transport and placing consists of two parts, whose partial optimums conflict. Thus, x_1 to x_7 stand for the "material" composition of the mix, and as this part is dominated by the cost of cement its partial optimum would require a mix with a minimum percentage of cement satisfying the previously defined and established technological criteria. In the second phase (stage) of the process, i.e. the placing of concrete defined by parameter x_8, the cost of placing is inversely proportional to the content of cement in the mix. Thus the partial optimum of this phase requires a mix with a maximum percentage of cement. Consequently, the complex optimum is to be found in a compromise between these requirements. the system of linear equations (presented here) can be solved, if each goal (right-hand side of the equation) is completely achieved; however, it is often impossible to do so as there are certain deviations caused by specific requirements (for example

the producer delivers enormous quantities of one grading). From the technological point of view these deviations may be acceptable, if they do not exceed the implied constraints, but they should be as low as possible. As goal programming (GP) makes it possible to achieve "reference values" with minimum deviations and as the system of priorities enables us to make compromises between the goals, we decided to apply this method in the numerical treatment of the described problem.

THE APPLICATION OF GOAL PROGRAMMING

Although goal-programming basically boils down to classical linear programming problems, it differs radically with regard to the way: a) objctive function and b) the set of constraints are formulated.

When formulating the objective function, we must give an estimated ideal value Y_r to which that function $F_r(x)$ sould tend. We must also determine priorities for realizing the given goals in terms of their effects on the solution of the given problem.

One characteristic feature in the consideration of a set of constraints is that each constraint is treated as a special goal to be achieved, while the deviation variables are alowed to exceed the implied constraints if this is acceptable from the technological point of view.

In the same way a GP task can be mathematically represented as an LP task by expressing all the objective functions as constraints:

$$a_{11}x_1+a_{12}x_2+\ldots\ldots + a_{1n}x_n + d_1^- - d_1^+ = b_1$$

$$a_{21}x_1+a_{22}x_2+ \ldots\ldots+ a_{2n}x_2 + d_2^- - d_2^+ = b_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1+a_{m2}x_2+\ldots\ldots + a_{mn}x_n + d_m^- - d_m^+ = b_m$$

$$c_{11}x_1+c_{12}x_2+ \ldots\ldots+ c_{1n}x_n + d_{m+1}^- - d_{m+1}^+ = Y_1$$

$$x_1 \ldots x_n;\ d_1^- \ldots d_{m+1}^-;\ d_1^+ \ldots d_{m+1}^+ \leq 0$$

and the objective function is the sum of the deviation variables, which should be minimized:

$$F = \min (d_1^- + d_1^+ \ldots\ldots + d_{m+1}^- + d_{m+1}^+)$$

In order to solve GP problems Lee developed an algorithm based on the simplex multi-criterional table and presented the GOALPG programme (Lee S., 1972) which was used as a sub-programme of a special programme developed for the numerical treatment of the problem of mixes. (Figure 2).

Successive solutions are obtained, (taking into account) the constraints:

$$x_1 < x_{1k}$$

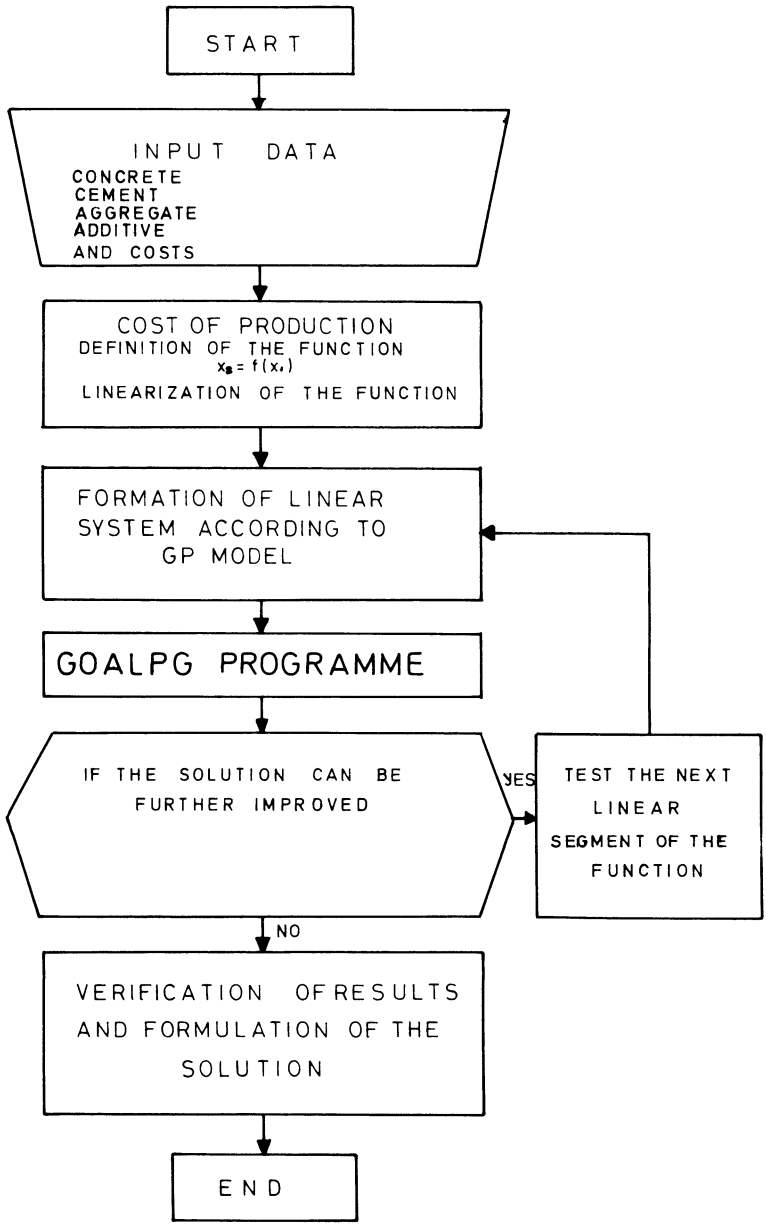


FIG. 3 FLOW DIAGRAM

and they corespond the points of the polygon. For this reason, in the testing of each line limits must be set.

CONCLUSION

The presented procedure can be employed to perform double optimization giving the recipe mix with optimal rheological properties and at the same time a minimum cost of placed concrete. Programme GOALPG can be used to define an optimum mix from the rheological stand point without taking cost into account, which is particularly suitable when preparing data for laboratory work.

References

ACI - Standard Recommended Practice for Selecting Proportions for Concrete (ACI 613)

Bonacci O., Mladineo N. (1981) Primjena linearnog programiranja za odredjivanje optimalnih smjesa u gradjevinarstvu, "Gradjevinar" 33/4, Zagreb

Lee S.,(1972) Goal Programming for Decision Analysis, Pennsylvania

Mladineo N., Krstulović P., Ramljak B., Primjena ciljnog programiranja za odredjivanje optimalnih mješavina u gradjevinarstvu, SYM-OP-IS' 82, Beograd

Nevil A.M.,(1976) Svojstva betona, Beograd

Powers T.C., (1968) The Properties of Fresh Concrete Wiley

Strok J., (1964) Teoria skladby betonovej zmesi, Bratislava

A COMPUTER PROGRAM FOR TIME SERIES ANALYSIS, MATRIX OPERATIONS, AND DATA MANIPULATION

Constantinos Georgiadis

SINTEF, The foundation of Scientific and Industrial Research at the Norwegian Institute of Technology, Trondheim, Norway.

INTRODUCTION

CGTSA is a Computer proGram developed for Time Series Analysis, matrix operations, and data manipulation. In an engineering office, research organization or university environment, time series analysis,matrix operations and data handling together with finite element analysis are in every day use, when dealing with dynamic analysis of structures subject to earthquake, wind or wave loading. Experience has shown us that a lot of wasted time and errors are the consequence of using a whole library of different programs, each doing a part of the job, converting data and manipulating files to communicate between them, or studying different users manuals.

This program has been developed with a definite point in mind. "To make easy and pleasant such work as: time series analysis, Fourier transforms, frequency domain analysis, statistical evaluation of data, matrix operations, analysis of experimental data, communication with finite element programs, and display of data and results in a desirable form". In other words the handling of these subjects with such a program should be as easy as basic arithmetic on a desk calculator.

The program is an interactive package. It communicates continuously with the user giving information on data properties, refreshing theoretical background, discussing options and limitations, avoiding in this manner the concept of using a "black-box", with the well known consequences. It gives an immediate view of the problem and new analysis paths and decisions can be made in the course of the run. Encountered problems can be investigated deeper, and data and results can be looked upon through different windows before making the final conclusions.

There is no need of a user manual or knowledge of a computer language in order to use the program. Users can become familiar with the program within a short time.

Many programs related to the subjects of this program have been reported. To mention some of them: Christiansen et al

(1981) have developed a time series analysis program related to wave theories for batch processing of the data, Sigbjørnsson et al (1982) presented a program developed for time series analysis related to structural identification problems for batch processing of the data, Hansten,O. (1982) reported a program for time series analysis of recorded data command oriented without an extensive dialog between program and user, Hartz et al (1982) reported an interactive version of a program for matrix analysis.

This paper will describe the basic functions of the program, the kind of dialog it establishes with the user, the visual displays of the results, applications, and future developments.

OBJECTIVES

The program development started at the University of Washington (Seattle Wash.) in 1980, and continue at SINTEF avd. 71 (Trondheim, Norway), during 1982-1983 (Georgiadis 1982) The program has been beveloped to serve the following purposes:

1. To be a basic tool for consulting and design engineering offices.

2. To serve as an educational and research tool.

3. To be used as a link and control program between the various structural analysis programs.

4. To use effectively the various input-output devices and visual display equipment.

5. To be independent of computer machine and facilities.

6. To be interactive, straight forward, and easy to use by persons with no computing background.

7. To be a self contained package, supply enough information to the user, and not to need a user manual.

8. To be fast, with small memory requirements, reliable, give enough warnings and error messages, to be stable and not to dump out the user when errors are encountered.

9. To possess a large flexibility in the analysis paths and data input-output operations.

10. To be updated and maintained easily.

DEVELOPMENT APPROACH

To fullfil the objectives mentioned above, the program development has been as follow:

Computer equipment

The following computer configuration has been considered for the program run:

1. Basic device for communication is an ordinary nongraphical VDU terminal. Digital plots can be obtained on the terminal.

2. Graphic terminals can be used for continuous plots.

3. The user can read in data from any peripheral device, card reader, disk reader, and can send output to various peripheral devices, line printers, graph plotters.

4. A CPU with memory 64K 32-bit words, is enough for the basic version of the program. A less powerful version can be implemented on smaller size CPU.

Software

Basic conditions of the algorithms used are:

1. Computer language is FORTRAN 77.

2. Additional library routines for mathematical calculations and digital plots of data have been avoided.

3. For the continuous plots the GPGS-F, and PLOT-10 routines have been used. Conversion to other systems is easy.

4. Programming efficiency has been followed for minimum storage requirements and fast response. All storage of data is in a blank common area. In order to reduce the response time, use of computer files has been avoided, except when the user wants to save data for later use, or the amount of manipulated data exceeds the memory limits.

5. A command oriented dialog has been chosen. The only requirement from the user is to enter correctly the name of the basic command consisting of four letters, and the program asks the user for the rest.

6. The user is continuously informed about memory status.

7. There is a large flexibility accepting inputs in various formats, or in unformated form, from various computer files and outputs of other programs.

8. The program is able to output data in forms readable by other programs.

9. The accuracy ot the software has been checked with other existing programs.

Subjects

The program can be used when dealing with the following subjects:

1. Time series anlysis in time or frequency domain.

2. Manipulation of data series recorded in situ or in experiments.

3. Basic matrix operations.

4. Statistical analysis of data.

5. Interface with structural analysis programs.

PROGRAM ARCHITECTURE

The basic architecture ot the program is shown in Fig.1. A table containing the names and addresses of the series and matrices is the central administration of the program. Time, data or frequency series and matrices are represented with numbers up to four digits as names. The number 10000 is added to the frequency series names to separate their names stored in the table, and matrix names are stored as negative numbers.

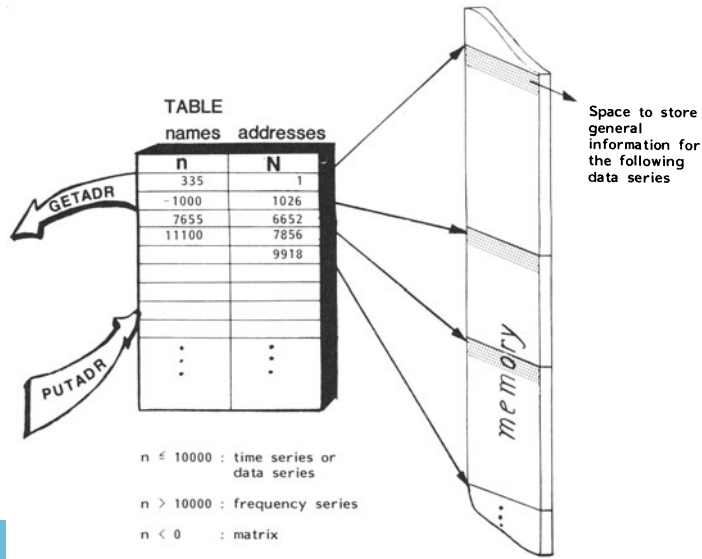


Figure 1. Memory management of the Program

In the memory (blank common) the series and matrices are stored consecutively as they are created, and their location can be found through the administration table. Additional memory locations are reserved at the begining of each Series or matrix in the memory, containing information for the rest of the stored data.

The way the program executes and interacts for each command is shown in Fig. 2. This consists basically of a comparison of the entered command with the table of possible commands, then the program calls to an appropriate control subroutine, interacts with the user for names of series and matrices involved, obtains addresses through the administrative table, creates new addresses and space for results, and calls the appropriate calculation subroutines.

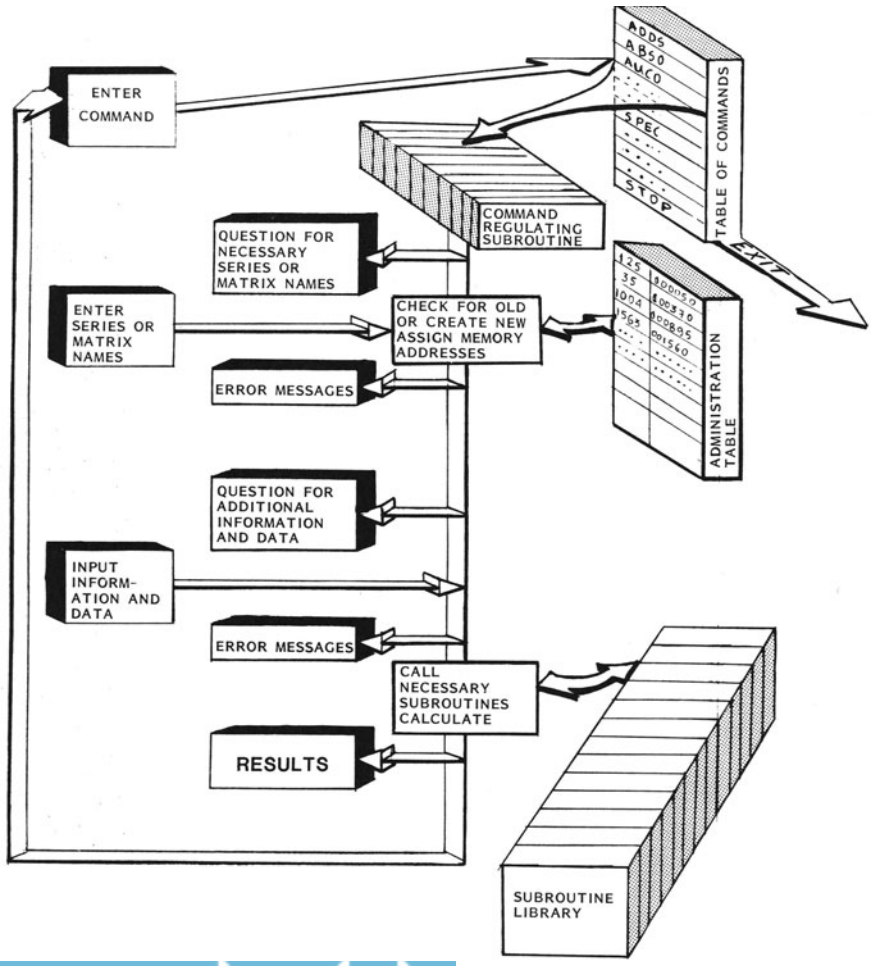


Figure 2. Program interactive execution sequence

PROGRAM FUNCTIONS

The basic functions of the program are shown in Fig. 3, and can be described generally as:

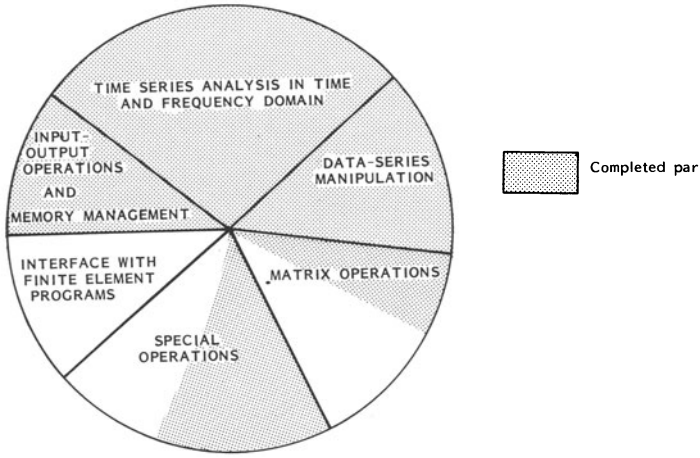


Figure 3. Program basic functions

Input-Output Operations

1. File manipulation, open, close, rewind.

2. Program memory management, allocate data in the memory, insert new or remove old data from memory.

3. Read and write data from and to any computer file, with flexibility in the format or in unformated form or compatible with outputs and inputs of other programs.

4. Print data on any peripheral device and terminal.

5. Plot data in digital form on regular CRTs, or continuous form on Tektronix terminals or continuous plotters.

6. Adjust format of printed or plotted data and display results of other programs in any desirable form.

Time Series Analysis in Time and Frequency Domain

1. Elementary operations in time or frequency domain, addition, substraction, scaling, multiplication, normalization, etc., of series.

2. Transformation from time to frequency or from frequency to time domain using Fast Fourier Transform techniques.

3. Correlation, cross-correlation, convolution of time series.

4. Spectra, cross-spectra and coherence calculations.

5. Filtering and windowing in time or frequency domain.

6. Statistics of time series, probability curves, maxima and minima distributions.

7. Integration and differentiation of series with finite difference methods or using Fourier components.

8. Time series simulation.

Data Series Manipulation

Statistics, editing, curve fitting to two dimensional (x, y) data.

Matrix Operations

1. Elementary matrix operations, addition, substraction, multiplication, transpose, editing, etc.

2. Solution of system of linear equations, matrix inversion.

3. Eigenvalue computations, mode shapes.

4. Time domain integration of system of dynamic equations of motion.

5. Frequency response calculations of dynamic systems.

6. Nonlinear vibrations.

Special Operations

1. Fatigue analysis, cycle counting.

2. Special manipulation or computations of data for structure identification problems.

3. Directional wave spectra prediction for limited fetch regions.

4. Prediction and simulation of various kinds of wave or wind spectra.

5. ARMA and linear filtering methods.

6. Monte Carlo simulation of time series with certain statistical properties, in order to be used as nodal loads in finite element programs.

7. Other operations for research purpose.

Interface with Finite Element Programs

1. Prepare input data.

2. Manipulate files and link between finite element programs.

3. Analyse the output data.

Program status

As can be seen from Fig. 3, the final version of the program is not ready. It is expected to be completed by summer 1983.

Table 1 shows a list and short description of the commands existing at the moment, as can be listed with the HELP request by the program. From these the large variety of analysis possibilities can be seen.

EXAMPLE OF USE

The best way to realize the efficiency of a program is to run it. Fig. 4 outlines some of the basic solution techniques of stochastic dynamics in time or frequency domain, which can be solved using the program along any of the solution paths shown in the figure. In Fig. 5, is shown an example of working with the program for some time series analysis. The way the program interacts and the form of the output and plotted data can be seen.

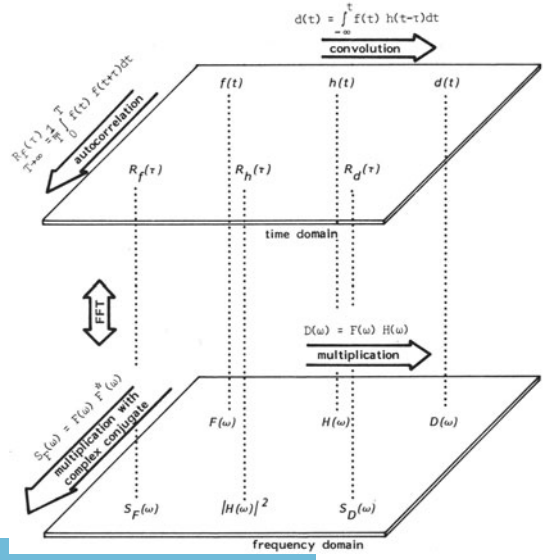


Figure 4. Basic solution procedures for problems of stochastic dynamics.

Table 1. Basic Program commands

********** BASIC OPERATIONS **********

In the program there are manipulated time frequency and X,Y coordinate series. Their names can be any number from one to four digits. Same names can used for time or frequency series A coordinate series is stored as time one.

The program always informs you about the necessary informations to be input.In case of error it will ask again .

When you getting the signal (***) enter a command from the following list.

When printing or plotting on the screen, alot of data ,each time a page is filled up you get the signal (&). If you hit return you get the next page of printed or plotted data. If you want to skip or backspace pages enter n or -n where n the number of pages to be skipped or backspaced.

The available ,for the moment, commands and their function are listed alphabetically in following:

&

ABSO Takes the absolute value of a time or freq. series.

ADDS Adds two time or frequency series.

ADDC Adds a constant to a time or frequency series.

AMPH Converts two time or freq. series containing real and imaginary parts in amplitude and phase angle series.

ARSP Computes the area under a spectrum assumed two sided.

AUCO Computes the autocorrelation function of a time series.

COHE Computes the coherence function of two time series.Before called the cross and autospectra should be computed

CONV Convolution of two time series .

CRCO Computes cross-correlation function of two time series

CRSP Computes cross-spectrum of two time series.

&

CYCL Cycle counting for fatigue analysis, gives the option of plotting cycles versus stress level.

DELE Deletes a time or frequency series from the memory.

DEMO Complex demodulation of time series.

DIFF Differentiates a time series.

DUPL Duplicates any part of a time or frequency series to a new series.Sampling interval can be changed.

EDIT To change certain values in a time or freq. series.

EDXY To change certain values in a coordinate series.

EXPO Raises a time or freq. series at a real or int. power.

&

FETC Computes fetch at certain direction for a closed region whose x,y coordinates are stored at a coordinate series

FFTR Computes the direct Fourier transform of a time series or the inverse Fourier transform of a freq. series. In both cases real and imaginary series are needed .

FILT Band passed filtering of frequency series.

FNGN To generate periodic or impulse functions.Special kinds of functions ,step,ramp,triangular,parabolic,sine,cos, half sine,half cos,or any desirable shape can be input.

FQRS Frequency response function for one degree of freedom.

FRID Integration or differantiation in frequency domain.

IMRS Impulse response function .

&

INIT Initializes devices,ploters, sets windows,char. sizes.

INTE Integrates a time or frequency series.

LDUB Constructs a series with double length and half time or frequency interval by linear interpolation.

LIST List of memory contains.

LOGR Takes the logarithm of a time or frequency series.

LPFL Low pass filtering of a time series in time domain.

LSFT Least square fit of polynomion up to 10th degree in x,y data stored in coordinate series.

MERG Merges a time or freq series to a bigger one additive.

MMMS Maximum, minimum, mean and st. deviation of a time ser.

MULT Multiplication of time or frequency series.

NLVR Response of one degree nonlinear system.

Table 1. (cont.)

OPEN Opens files with different properties.

PLOT Digital or continuous plot of time or frequency series To get plots in different scale duplicate only a part of the series to another one.

PLLG Logarithmic plot of x,y data stored in coordin. series

PLXY Plot of x,y data stored in coordinate series. A line if desired can be plotted between the data points and a line of a least square fitted polynomion up to 10th degree.

PXYC Prints a coordinate series. The series can be printed in any peripheral device, after it has been connected with an OPEN statement.

POCA Converts a coordinate series from polar to cartesian.

&

PRIN Prints a time or frequency series. The series can be printed in any peripheral device, after it has been connected with an OPEN statement.

PRRP Probability function of random process.

READ Reads a time or frequency series. The series can be read in from the terminal or from any computer file after the file has been connected via an OPEN command. The data can be read in formated or unformatted form, the format can be specified from the user.

REFE Presents informations about the program and interesting references on subjects the program deals with.

REWI Rewinds a file.

RMVL Removes the mean or a line component from a time series data.

&

RXYC Reads a coordinate series with x,y data. The series can be read in from the terminal or any computer file provided it has been connected via an OPEN statement. The data can be read in formatted or unformatted form. In case of formatted data use FORMAT(10X,F20.0). In case of unformatted data, should store first X and then Y coordinates in one record.

SCAL Scales a time or freq. series by a constant.

SHIF Shifts a time series by certain time interval adding zeros in the end.

SIMU Simulates a time series from a spectrum.

SINC Forms a time series by superimposing sine components at different phases.

& _

SMOO Smooths a time or freq. series using a moving average.

SMSP Smooths spectra with various filters.

SPEC Autospectrum calculation of a time series. Smoothing with a Daniel filter. The length and number of the filter applications can vary.

STOP Terminates program execution.

SUBS Substracts two time or frequency series.

STPL Fits with least squares a plane in stress data time series, and computes axial forces, bending moments, angle of stress plane, as new time series.

SYMA From an existing series constructs a symmetric or antisymmetric one around the y axis.

&

TAPW Time series is tapered with various windows.

WAVS Forms wave spectrum Bretsneider, Jonswap, ...

WIND Forms wind spectra Davenport, Harris, ...

WNCI Forms a white noise time series.

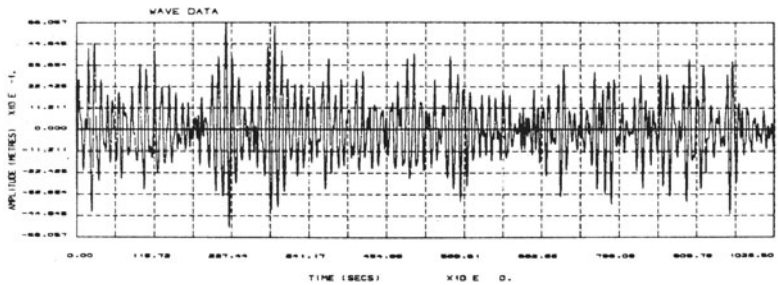
WRIT Writes a time, frequency, coordinate series in a file in binary form. The file should have been previously opened with an OPEN command.

ZERO Forms a zero time or freq. series.

& _

e

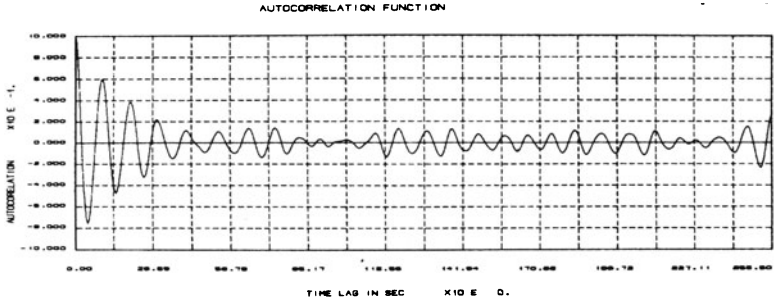
```
***OPEN                          AVAILABLE MEMORY : 50000  OCUPIED MEMORY :      0

OPEN FILE ENTERING FOLLOWING
 ENTER UNIT NUMBER :  10
 ENTER FILE NAME   :  MAINDATA:D
 ENTER STATUS  OLD,NEW,UNKNOWN :  O
 ENTER ACCESS W write, R read,RW readandwrite default SEQUENTIAL :
 ENTER FORM FORMATTED OR UNFORMATTED :  U
SO YOU WANT TO OPEN A FILE AS FOLLOWS :
FILE UNIT :  TAPE 10
FILE NAME :  MAINDATA:D
STATUS    :  OLD
ACCESS    :  SEQUENTIAL
FORM      :  UNFORMATTED
                          Y ES OR  N O :  Y
FILE HAS BEEN OPENED O.K.

***                              AVAILABLE MEMORY : 50000  OCUPIED MEMORY :      0

***READ                          AVAILABLE MEMORY : 49999  OCUPIED MEMORY :      1
 ENTER  T IME OR  F REQUENCY DOMAIN :  T
 ENTER SERIES NAME :  1
 ENTER NUMBER OF TIME OR FREQUENCY POINTS :  2048
 ENTER TIME OR FREQUENCY INTERVAL :  .5
 ENTER BEGINING TIME OR FREQUENCY :
 ENTER DEVICE ID, 0=TERMINAL OR UNIT NUMBER :  10
 ENTER F ORMATTED OR U NFORMATTED INPUT :  U
 ENTER SERIES IDENTIFICATION NUMBER :  1
 ENTER SERIES VERSION NUMBER        :  1


  TIME SERIES READ FROM FILE :      10
  TIME SERIES NUMBER         :       1 :      1
  IDENTIFICATION HEADING     :  *.
                                *.
  NUMBER OF POINTS           :    2048
  TIME INTERVAL              :       .50000

***RMVL                          AVAILABLE MEMORY : 47949  OCUPIED MEMORY :   2051
 ENTER TIME SERIES NAME :  1

 OPTION TO REMOVE MEAN OR BEST LEAST SQUARE FITTED LINE
 ENTER  M EAN OR  L INE :  M

***PLOT                          AVAILABLE MEMORY : 47949  OCUPIED MEMORY :   2051
 YOU WANT  D IGITAL OR  C ONTINUOUS PLOTS :  C
 ENTER  T IME OR  F REQUENCY DOMAIN :  T
 ENTER SERIES NAME :  1
 DO YOU WANT R EGULAR OR L OG PLOT :

ENTER BELOW LABEL OF GRAPH  60 LETTERS MAXIMUM
WAVE DATA

ENTER BELOW LABEL OF VERTICAL AXIS 20 LETTERS
AMPLITUDE (METRES)

ENTER BELOW LABEL OF HORIZONTAL AXIS 20 LETTERS
TIME (SECS)
 DO YOU WANT GRID TO APPEAR  Y ES OR N O :  Y
```




```
***AUCO                          AVAILABLE MEMORY : 46922  OCUPIED MEMORY :   3078
 ENTER TIME SERIES NAME :  1
 ENTER NAME OF SERIES WITH AUTOCORRELATION :  11
 ENTER NUMBER OF LAGS DEFAULT =  512  :
 DO YOU WANT NEGATIVE LAGS TOO Y ES OR N O :

DO YOU WANT AUTOCOVARIANCE ...ACOV
     OR    AUTOCORRELATION ...ACOR  :  ACOR
```

Figure 5. Example of using the program for time series analysis (read, plot series, autocorrelation, FFT, probability curves, digital plots)

```
***PLOT                          AVAILABLE MEMORY : 46408  OCUPIED MEMORY :  3592
 YOU WANT  D IGITAL OR  C ONTINUOUS PLOTS :  C
 ENTER  T IME OR  F REQUENCY DOMAIN :  T
 ENTER SERIES NAME :  11
 DO YOU WANT R EGULAR OR L OG PLOT :  R

ENTER BELOW LABEL OF GRAPH  60 LETTERS MAXIMUM
AUTOCORRELATION FUNCTION

ENTER BELOW LABEL OF VERTICAL AXIS 20 LETTERS
AUTOCORELATION

ENTER BELOW LABEL OF HORIZONTAL AXIS 20 LETTERS
TIME LAG IN SEC
 DO YOU WANT GRID TO APPEAR  Y ES OR N O :  Y
***                              AVAILABLE MEMORY : 46408  OCUPIED MEMORY :  3592
```




```
***ZERO                          AVAILABLE MEMORY : 47949  OCUPIED MEMORY :  2051
 ENTER  T IME OR  F REQUENCY DOMAIN :  T
 ENTER SERIES NAME :  2
 ENTER NUMBER OF TIME OR FREQUENCY POINTS :  2048
 ENTER TIME OR FREQUENCY INTERVAL :  .5

***FFTR                          AVAILABLE MEMORY : 45899  OCUPIED MEMORY :  4101
 ENTER  D IRECT OR  I NVERSE :  D
 DO YOU WANT TRANSFORMED SERIES H ALF OR F ULL LENGTH :

AVAILABLE FFT  S-T  SANDE-TUKEY, C-L COOLEY-LEWIS
 CHOOSE ONE  DEFAULT S-T :
 ENTER REAL TIME SERIES NAME :  1
 ENTER IMAG TIME SERIES NAME :  2
 ENTER REAL FREQUENCY SERIES NAME :  1
 ENTER IMAG FREQUENCY SERIES NAME :  2

 LENGTH OF REAL SERIES :  2048 LENGTH OF IMAG SERIES :  2048
 ENTER TRANSFORMED LENGTH  DEFAULT  2048 :

***PLOT                          AVAILABLE MEMORY : 43845  OCUPIED MEMORY :  6155
 YOU WANT  D IGITAL OR  C ONTINUOUS PLOTS :  C
 ENTER  T IME OR  F REQUENCY DOMAIN :  F
 ENTER SERIES NAME :  1
 DO YOU WANT R EGULAR OR L OG PLOT :

ENTER BELOW LABEL OF GRAPH  60 LETTERS MAXIMUM


ENTER BELOW LABEL OF VERTICAL AXIS 20 LETTERS


ENTER BELOW LABEL OF HORIZONTAL AXIS 20 LETTERS
***                              AVAILABLE MEMORY : 43845  OCUPIED MEMORY :  6155
```

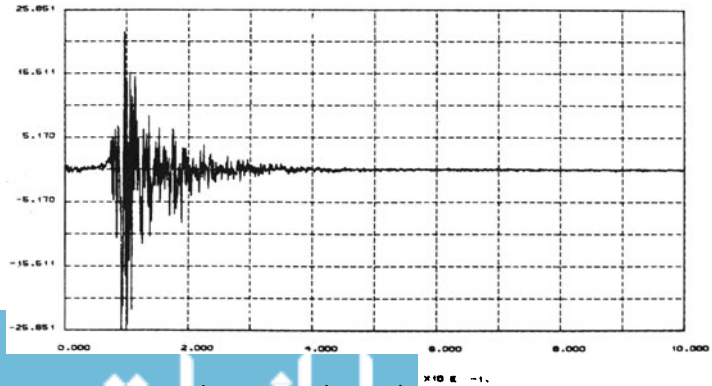
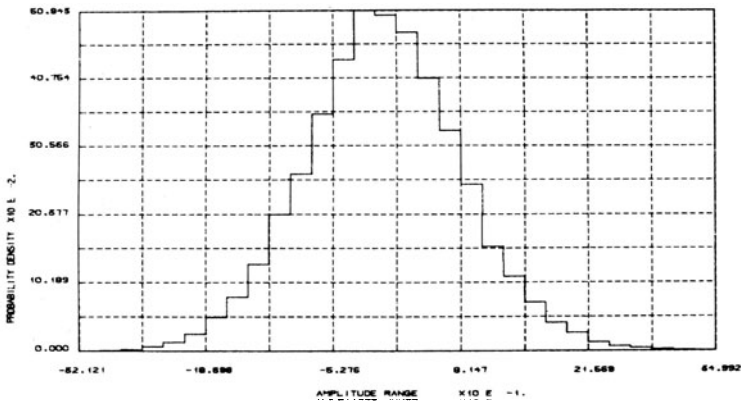


Figure 5 (cont.)

```
***PRRP                          AVAILABLE MEMORY : 46922  OCUPIED MEMORY :  3078
 ENTER TIME SERIES NAME :  1
 ENTER NUMBER OF AMPLITUDE LEVELS :  20
 DO YOU WANT PLOT OF PROB. CURVES Y ES OR N O :  Y
 YOU WANT  D IGITAL OR  C ONTINUOUS PLOTS :  C

ENTER BELOW LABEL OF GRAPH  60 LETTERS MAXIMUM
WAVE DATA

      PROBABILITY DENSITY RESULTS
      ***************************

      AMPLITUDE FROM :   -.5096E+01  TO :   -.4561E+01   PROBABILITY :   .002204
      AMPLITUDE FROM :   -.4561E+01  TO :   -.4026E+01   PROBABILITY :   .006350
      AMPLITUDE FROM :   -.4026E+01  TO :   -.3491E+01   PROBABILITY :   .013088
      AMPLITUDE FROM :   -.3491E+01  TO :   -.2956E+01   PROBABILITY :   .031075
      AMPLITUDE FROM :   -.2956E+01  TO :   -.2421E+01   PROBABILITY :   .043202
      AMPLITUDE FROM :   -.2421E+01  TO :   -.1886E+01   PROBABILITY :   .072855
      AMPLITUDE FROM :   -.1886E+01  TO :   -.1351E+01   PROBABILITY :   .156529
      AMPLITUDE FROM :   -.1351E+01  TO :   -.8155E+00   PROBABILITY :   .210662
      AMPLITUDE FROM :   -.8155E+00  TO :   -.2804E+00   PROBABILITY :   .284824
      AMPLITUDE FROM :   -.2804E+00  TO :    .2547E+00   PROBABILITY :   .277360
      AMPLITUDE FROM :    .2547E+00  TO :    .7898E+00   PROBABILITY :   .234254
      AMPLITUDE FROM :    .7898E+00  TO :    .1325E+01   PROBABILITY :   .200706
      AMPLITUDE FROM :    .1325E+01  TO :    .1860E+01   PROBABILITY :   .129392
      AMPLITUDE FROM :    .1860E+01  TO :    .2395E+01   PROBABILITY :   .088693
      AMPLITUDE FROM :    .2395E+01  TO :    .2930E+01   PROBABILITY :   .056596
      AMPLITUDE FROM :    .2930E+01  TO :    .3465E+01   PROBABILITY :   .026388
      AMPLITUDE FROM :    .3465E+01  TO :    .4000E+01   PROBABILITY :   .021246
      AMPLITUDE FROM :    .4000E+01  TO :    .4535E+01   PROBABILITY :   .007552
&     AMPLITUDE FROM :    .4535E+01  TO :    .5071E+01   PROBABILITY :   .002158

AS COMPUTED FROM PROBAB. CURVE :
 MEAN VALUE :   -.1773E-03   ST. DEVIAT. :    .8198E+00
 THIRD MOMENT:   .4168E-01   KYRTOSSIS   :    .3217E+01
 DO YOU WANT PLOT OF PROBAB. CURVE Y ES OR N O :
```




```
DO YOU WANT PLOT OF NORM. SCALE Y ES OR N O :
```

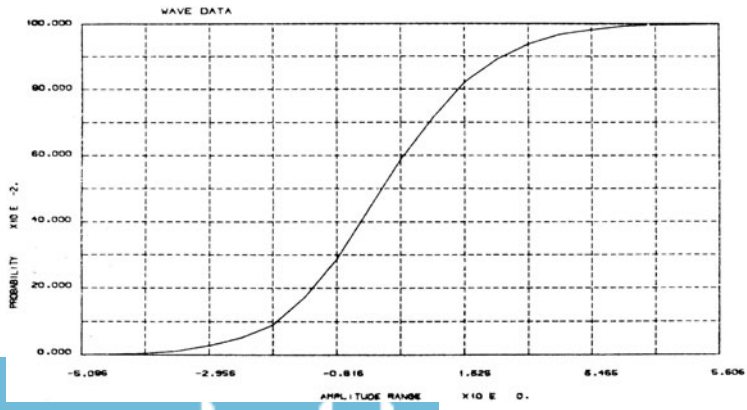
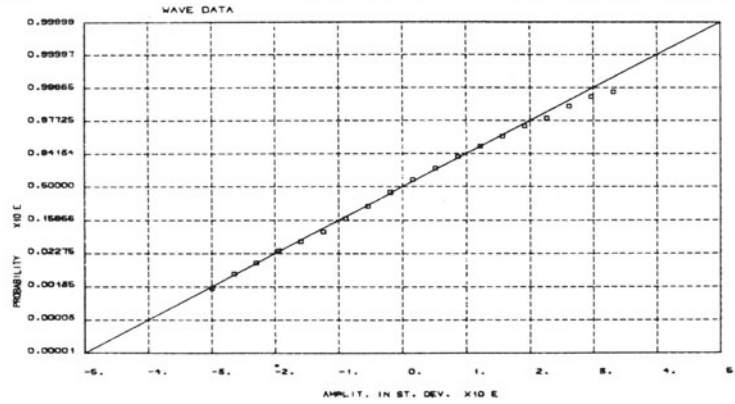


Figure 5 (cont.)

```
***                          AVAILABLE MEMORY : 46922  OCUPIED MEMORY :  3078
```




```
***PLOT                      AVAILABLE MEMORY : 41791  OCUPIED MEMORY :  8209
 YOU WANT  D IGITAL OR  C ONTINUOUS PLOTS :  D
 ENTER  T IME OR  F REQUENCY DOMAIN :  F
 ENTER SERIES NAME :  1
 DO YOU WANT R EGULAR OR L OG PLOT :
 ENTER DEVICE ID, 0=TERMINAL OR UNIT NUMBER :

******  PLOT OF POSITIVE ORDINATES  ******
 .305E+00I---*-----------------------------------------------I
          I   *                                               I
          I   *                                               I
          I   *                                               I
 .244E+00I---*-----------------------------------------------I
          I   *                                               I
          I   *                                               I
          I   *                                               I
 .183E+00I---*-----------------------------------------------I
          I   *                                               I
          I   **                                              I
          I   **                                              I
 .122E+00I---**----------------------------------------------I
          I  ****                                             I
          I  ****                                             I
          I  ******                                           I
 .609E-01I--******---------------------------------------------I
          I  ********                                         I
          I  *********                                        I
          I* ************                                     I
          ----+----+----+----+----+----+----+----+----+----+
             .10       .30       .50       .70       .90      1.00

***                          AVAILABLE MEMORY : 41791  OCUPIED MEMORY :  8209


***LIST                      AVAILABLE MEMORY : 35637  OCUPIED MEMORY : 14363

  1          T.S.    1    START    1      END 2051     LENGTH 2048
  2          T.S.    2    START 2051      END 4101     LENGTH 2048
  3          F.S.    1    START 4101      END 5128     LENGTH 1025
  4          F.S.    2    START 5128      END 6155     LENGTH 1025
  5          T.S.    3    START 6155      END 7182     LENGTH 1025
  6          T.S.    4    START 7182      END 8209     LENGTH 1025
  7          T.S.   11    START 8209      END10259     LENGTH 2048
  8          T.S.   12    START10259      END12309     LENGTH 2048
  9          F.S.   12    START12309      END13336     LENGTH 1025
 10          F.S.   13    START13336      END14363     LENGTH 1025
***                          AVAILABLE MEMORY : 35637  OCUPIED MEMORY : 14363
```

Figure 5 (cont.)

APPLICATIONS, CONCLUSIONS

The efficiency of the program presented can be seen through its applications. The visual effects and the flexibility it possesses are of great value in a major part of the design and analysis of structures. The use of the program as a research tool for structural identification problems has shown the ability of the program to take care of a large number of data and analysis methods. The program has been used to analyse response data of offshore structures. Used in conjuction with finite element programs a complete analysis and design package is obtained. In designing structures under dynamic loading the simulation methods for time series, and the prediction of loading from environmental conditions, such as directional wave spectra or wind spectra, are very helpful.

As an educational tool the program can be used for many engineering classes, like: matrix structural analysis, dynamics of structures, stochastic and spectral analysis, wave and wind analysis, etc. The student can have an immediate simulation of the studied problems in the computer. Problems and methods of approaching structural response problems, as the one described generally in Fig. 4, can be solved and simulated easily.

The continuous information and warning to the user, and the simplicity of the program, help to reduce errors, and make it pleasant to use.

The program has been implemented on VAX11, and ND-100, ND-500. Its implementation on other computers is not problematic.

REFERENCES

1. Christiansen, D., Heavner, J. (1981) "NSAP Numerical Spectral Analysis Program", University of Washington, Seattle Washington.
2. Georgiadis, C. (1982) "CGTSA A Computer Program for Time Series Analysis-Working Manual". To be published as SINTEF Report, Trondheim, Norway.
3. Hansteen, O.E. (1982) "Users Manual to SANTO - A computer Program to Analyse Measured Time Series", Norges Geotekniske Institutt.
4. Hartz, B.J., Lary J.A., Dolan, J.D. (1981) "S.M.I.S. Interactive Symbolic Matrix System", University of Washington, Seattle Washington.
5. Sigbjørnsson, R., Hoen, C. and Hage, A. (1982) "STARTIMES A Computer Program for Statistical Analysis of Random Series with Special References to Structural Dynamics", SINTEF Report STF71 A82011, Trondheim, Norway.

NUMERICAL ANALYSIS OF ENERGY DISSIPATING DEVICES BY AN EXPLICIT TIME INTEGRATION PROCEDURE

J.M. Goicolea, K.R.F. Andrews, G.L. England

King's College, London

INTRODUCTION

The recent developments in fields such as Nuclear Energy or Transportation and the concern over safety in accidents have generated a great deal of interest in the last few years for the analysis of Energy dissipating devices. By this term we understand mechanisms that absorb the Kinetic Energy at collision in an irreversible manner, and collapse in a predictable fashion, should an unintentional impact take place.

One such class of mechanisms which possess a high Energy absorption capability (Johnson et al., 1978) are circular tubes or tube assemblies subjected to axial crumpling. Some experimental work has been done to assess the collapse modes and Energy absorption characteristics (e.g. Ghani, 1982), but an accurate theoretical modelling of their behaviour is still lacking. Such models would need to incorporate dynamic analysis, large strains and displacements, and non-linear constitutive laws, making their study extremely challenging. The characteristics of the problem point to a numerical solution with spatial discretization of the continuum and direct time integration.

Finite Element or Finite Difference methods which use time-marching integration can be classified roughly into two types: explicit methods, which use the values of accelerations and velocities at the present and previous instants (n, n-1, n-2...) to obtain the new displacements at time n+1, and implicit methods, which use also the values at time n+1, and therefore need to solve an implicit equation between forces and displacements at that instant, which is usually done by inverting a stiffness matrix.

Although stability requirements impose smaller time-steps for explicit integration methods, several features make them more attractive for our case: firstly for impact loading it is important to consider stress-wave propagation effects, which would lead us to take small time-steps in any case regardless of stability considerations; The strong material and geometric non-linearities also impose small time-steps if we are to avoid iteration; therefore the reassembling and inversion of the stiffness matrix at each step would make implicit methods uneconomical. Finally explicit schemes allow with great ease the direct implementation of complex constitutive laws.

A choice may also be made between the use of Finite Difference Methods (FDM) or Finite Element Methods (FEM). Although FDM lack the versatility of FEM with regard to higher order elements or shell elements, they provide an efficient and straightforward algorithm (Marti, 1981) for simple elements such as the constant strain triangle used here. Furthermore for this case both formulations can be shown to be equivalent (Kunar et al., 1981).

A Finite Difference numerical integration procedure for the non-linear dynamic analysis of deformable continua is proposed here, using an explicit central difference time-marching scheme. The algorithms have been implemented in a computer code which can tackle a wide range of two-dimensional (plane stress or plane strain) and axisymmetric problems. In particular it is applied to the dynamic analysis of elastoplastic circular tubes submitted to gross axial deformation.

THEORY

General

The continuum is discretized into a mesh of triangular cells. A diagonal mass matrix is used, the masses being lumped at the nodes in the cell corners. Besides the simplification of allowing the equations of motion to be uncoupled, the diagonal mass matrix idealization produces a distortion in the frequency spectrum of opposite sign to that caused by the explicit integration scheme, thus compensating each other (Krieg et al. 1973).

Each computational cycle begins with the relative nodal displacements being used to compute the strains in the cells; these are fed into the constitutive law, yielding the stresses, which are then integrated around the nodes and added to the exterior and body forces to obtain the equivalent nodal forces. The application of the momentum equations will produce the new displacements, and the cycle

is repeated the next time step. An updated Lagrangian procedure is used, where the nodal coordinates are updated after each time-step, and the incremental strains during each step are assumed to be small.

Constitutive equations

The equations are formulated in an incremental fashion. Firstly the strain rates are obtained through linear interpolation from the nodal velocities in each cell:

$$\{\dot{\varepsilon}_{ij}\} = [B]\{\dot{u}_i\} \tag{1}$$

where
- ε_{ij} strain tensor
- u_i nodal displacements
- $[B]$ interpolation matrix
- dots represent material time derivatives

The stress rates are then obtained through the application of the desired constitutive law; this law can be arbitrary, as long as it can be formulated explicitly, i.e.

$$\overset{\nabla}{\sigma}_{ij} = F_{ij}\,(\varepsilon_{lm}, \dot{\varepsilon}_{lm}, \sigma_{lm}, \{K\}) \tag{2}$$

where
- σ_{lm} stress tensor
- F_{ij} arbitrary functions with symmetric tensor characteristics
- $\{K\}$ set of material properties

$\overset{\nabla}{\sigma}_{ij}$ stands for the Jaumann co-rotational stress rate:

$$\overset{\nabla}{\sigma}_{ij} = \dot{\sigma}_{ij} + \sigma_{ip}\Omega_{pj} + \sigma_{jp}\Omega_{pi} \tag{3}$$

where Ω_{ij} is the skew-symmetric component of the velocity gradient tensor:

$$\Omega_{ij} = \tfrac{1}{2}(\dot{u}_{i,j} - \dot{u}_{j,i}) \tag{4}$$

A similar correction term to that used in equation (3) must be applied to obtain the spin-invariant rates of any first or second order tensor linked to the material points whose value we would wish to monitor in the original fixed frame. This would apply, for instance, to the strain tensor, or to a set of anisotropic material properties.

An isotropic elastoplastic constitutive law has been implemented. The elastic constitutive relations are:

$$\overset{\nabla}{\sigma}_{ij} = \lambda\dot{\varepsilon}_{kk}\delta_{ij} + 2\mu\dot{\varepsilon}_{ij} \tag{5}$$

where
- λ, μ Lame constants
- δ_{ij} Kronecker delta

A Von Mises yield criterion with isotropic hardening is used. The deviatoric stresses are scaled down to remain inside the current yield surface:

$$s_{ij}^{n+1} = \min\left[\frac{\tau_y}{\tau_{eff}}, 1\right](s_{ij}^{n} + \dot{s}_{ij}^{n+\frac{1}{2}}\Delta t) \qquad (6)$$

$$s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{\alpha\alpha}$$

where τ_y = value of τ_{eff} at yield.

$$\tau_{eff} = \left(\frac{3}{2}s_{ij}.s_{ij}\right)^{\frac{1}{2}}$$

The tangent plastic hardening modulus H' is set so the material behaviour adjusts to a given uniaxial stress-strain curve, $\sigma = \sigma(\varepsilon)$

$$\frac{1}{H'} = \frac{1}{H} - \frac{1}{E} \qquad (7)$$

where $H = \frac{d\sigma}{d\varepsilon}$ (elastoplastic modulus)

Momentum equations

The equations governing the balance of momentum are:

$$\sigma_{ij,j} + \rho(f_i - \ddot{u}_i) = 0 \qquad (8)$$

where ρ mass density

f_i body forces for unit mass

Integrating on a small space domain V around each node and transforming the stress integral according to the Gauss-Ostrogradsky theorem:

$$\ddot{u}_i = \frac{\int_S \sigma_{ij} n_j \, ds + Mf_i + F_i}{M} \qquad (9)$$

where F_i resultant of external forces

S surface bounding domain V

M $\int \rho dV$

n_j exterior normal to S

Finally, the accelerations from equation (9) are used to compute velocities and displacements with a central difference scheme:

$$\dot{u}_i^{n+\frac{1}{2}} = \dot{u}_i^{n-\frac{1}{2}} + \ddot{u}_i^{n}.\Delta t \qquad (10)$$

$$u_i^{n+1} = u_i^{n} + \dot{u}_i^{n+\frac{1}{2}}\Delta t \qquad (11)$$

Rayleigh viscous damping, with mass and stiffness-proportional terms, can be included.

The stability of the integration process is governed by the

Courant condition, i.e. stress waves must not be allowed to travel between cells within one computational cycle. This way the deformation in different individual cells will be uncoupled, assumption we are making in our explicit algorithm. The time-step size is given by:

$$\Delta t = C \frac{h_{min}}{V_p} \tag{12}$$

where C constant < 1
h_{min} minimum height in any triangular cell
V_p highest speed of stress waves

For the elastoplastic model V_p corresponds to the elastic longitudinal waves:

$$V_p = \left(\frac{\lambda + 2\mu}{\rho}\right)^{\frac{1}{2}} \tag{13}$$

Mixed discretization

With certain discretizations and in particular with plane strain triangles, Finite Element or Finite Difference solutions for plastic incompressible flow exhibit a too stiff response, due to an artificial overconstraining of the mesh when enforcing the local incompressibility condition (Nagtegaal et al., 1974).

To avoid this problem the number of degrees of freedom per element of the mesh must exceed the number of incompressibility constraints. As increasing the number of degrees of freedom can result in a singular stiffness matrix, this is normally achieved in FEM with reduced integration techniques, which diminish the number of integration stations and hence the incompressibility constraints. With an explicit algorithm there is no problem increasing the number of degrees of freedom, as the stiffness matrix is not inverted, but care must be taken to prevent 'hourglassing' modes from developing (i.e. deformation modes which produce zero stresses).

A mixed discretization procedure (Marti et al., 1982) which allows plastic incompressible flow without generating unwanted hourglassing has been adopted. Different discretizations are used for the isotropic and deviatoric parts of the stress and strain tensors: triangular cells are used for the deviatoric components, while the volumetric part uses the quadrilaterals formed by each two adjacent triangles. For this purpose volumetric strains are computed in each quadrilateral, N, averaging for the two triangular cells, N_1 and N_2, that compose it:

$$\varepsilon_{\alpha\alpha}^{N} = \frac{\varepsilon_{\alpha\alpha}^{N1} V_{N1} + \varepsilon_{\alpha\alpha}^{N2} V_{N2}}{V_{N1} + V_{N2}} \qquad (14)$$

This reduces the number of constraints due to the incompressibility condition, $\varepsilon_{\alpha\alpha} = 0$, from one per triangular cell to one for each pair of cells.

To test this procedure a Prandtl punch test has been modelled. The theoretical collapse load for a perfectly plastic material in plane strain is known through slip-line theory:

$$p = (2 + \Pi)\, C_u \qquad (15)$$

where p pressure at collapse
C_u shear strength

The mesh used can be seen in figure 1 where the length of the die is represented by a thick line. The load-displacement curve obtained (figure 2) shows a good adjustment to the theoretical limit load, compared to a case using ordinary constant strain triangles without mixed discretization, which shows no limit load at all.

APPLICATION: AXIAL COLLAPSE OF TUBES

Axial collapse of circular tubes can occur under two distinct deformation modes: Concertina mode, involving multiple bellows-type axisymmetric folds; or diamond lobe mode, which comprises several cyclically symmetrical triangular folds around the circumference of the tube. Both deformation modes can develop either simultaneously along the entire length of the tube, or through sequential folding starting at one end. The particular collapse mechanism followed in each case can be related experimentally to the ratios t/D and L/D (t tube thickness, D inner diameter, L, length).

It has been suggested (Ghani, 1982) that the most efficient energy absorbers are those which collapse sequentially under a concertina mechanism. In this case the Energy is dissipated progressively with the plastic deformation involved in the formation of each consecutive fold, and a fairly stable average collapse load is achieved.

In this paper we are concerned with the modelling of the concertina sequential collapse mechanisms. We have chosen a tube geometry in which the ratios t/D = 0.067 and L/D = 0.67 suggest the likelihood of this mechanism occurring (figure

3). An elastoplastic material with the following properties has been used:

Young's modulus	$E = 10^5$ N/mm^2
Poisson's ratio	$\nu = 0.3$
Uniaxial yield stress	$\tau_y^o = 10^3$ N/mm^2
Mass density	$\rho = 1.0$ kg/mm^3

Two examples are presented, one using a perfectly plastic material and another with strain-hardening characteristics following a bilinear stress-strain law (figure 4). In both cases the ends of the tubes have been restrained from rotating and a single fold develops, due to the small L/D ratio of the model used. It can be nonetheless a good estimate for the sequential collapse mechanism, as further progressive collapse in a longer tube would involve the same deformation repeatedly for each subsequent fold. No initial imperfections were considered.

Figure 5 shows the undeformed and deformed shapes at various stages for the perfectly plastic case (both cases give similar deformation patterns), while the load-displacement curves obtained can be seen in figures 6 and 7. The numerical results can be compared with the simplified theory proposed by Alexander (1960), which results in an expression for the mean collapse load:

$$\bar{P} = C\ \tau_y^o t\ \sqrt{Dt} \qquad \text{where } C \simeq 6 \qquad (16)$$

Eq. (16) yields a value of $P = 3.72 \times 10^5$ N, which compares rather well with the value of 4.3×10^5 obtained for the load in the stable post-buckling range in the perfectly plastic analysis. Alexander's theory does not include strain-hardening effects, so for this other case we shall consider an extension to Alexander's theory proposed by Ghani (1982):

$$\bar{P} = \tau_y^o \left\{\frac{\pi t^2}{2}\left(\frac{\pi D}{2h} + 1\right) + \pi h t\right\} + \frac{\pi H t}{6}\left\{\frac{3\pi^2 D t^2}{32\, h^2} + \frac{t^2}{h} + \frac{4h^2}{D+t}\right\} \qquad (17)$$

where $h = \left(\frac{\pi}{2\sqrt{3}}\ Dt\right)^{\frac{1}{2}}$

H = elastoplastic modulus

Eq. (17) yields a value of $P = 11.34 \times 10^5$ N, agreeing very well with the numerical results which show an average post-buckling collapse load of 11.5×10^5 N

ACKNOWLEDGEMENT

The authors wish to thank Dr. J. Marti of Principia Mechanica Ltd. for his invaluable help and advice.

REFERENCES

Alexander, J.M. (1960) An approximate analysis of the collapse of thin cylindrical shells under axial loading, Quart. J. Appl. Math., XIII pt. 1, pp. 10-15.

Ghani, E. (1982) Static and dynamic collapse behaviour of cylindrical tubes and tube assemblies, Ph.D. thesis, U. of London, King's College.

Johnson, W. and Reid, S.R. (1978) Metallic energy dissipating systems, Applied Mechanics Reviews, Vol. 31 pp. 277-288.

Krieg, R.D. and Key, S.W. (1973) Transient shell response by numerical time integration, Int. J. Num. Meth. Eng. Vol.7, pp. 273-286.

Kunar, R.R. and Minowa, N. (1981) A comparison between Finite Element and Finite Difference explicit formulations for triangular and quadrilateral plane strain elements, Int. J. Num. Anal. Methods in Geomechanics, Vol. 5. No.2 pp. 217-224.

Marti, J. (1981) PR3D - Programmer's guide, Principia Mechanica Ltd., London, Report No. PR-TN-4.

Marti, J. and Cundall, P.A. (1982) Mixed discretization procedure for accurate modelling of plastic collapse, Int. J. Num. Anal. Methods in Geomechanics Vol. 6. No. 1 pp. 129-139.

Nagtegaal, J.C., Parks, D.M. and Rice, J.R. (1974) On numerically accurate Finite Element solutions in the fully plastic range, Comp. Meth. Appl. Mechanics and Engineering, Vol. 4. pp. 153-174.

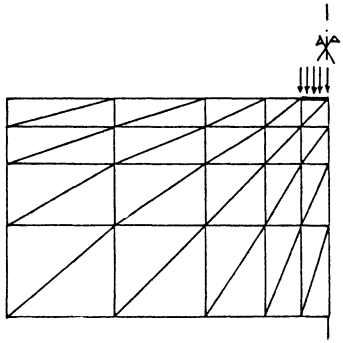


Figure 1 Punch Test Mesh.

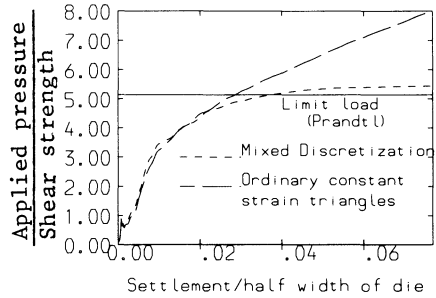


Figure 2 Punch Test Load-displacement Curve.

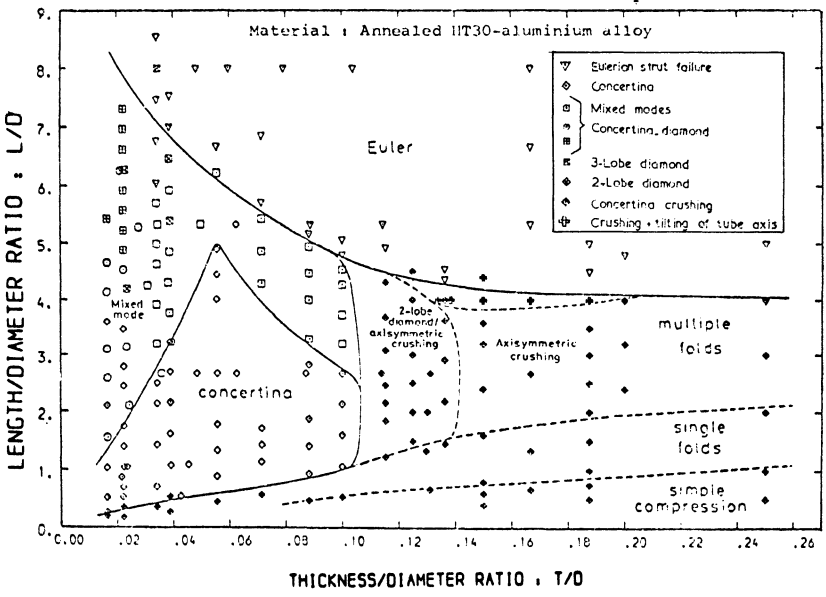


Figure 3 Classification Chart of Axial Collapse Modes in Cylindrical Tubes (from Ghani (1982)).

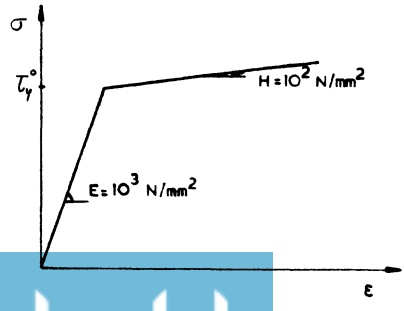


Figure 4 Uniaxial bilinear stress-strain law.

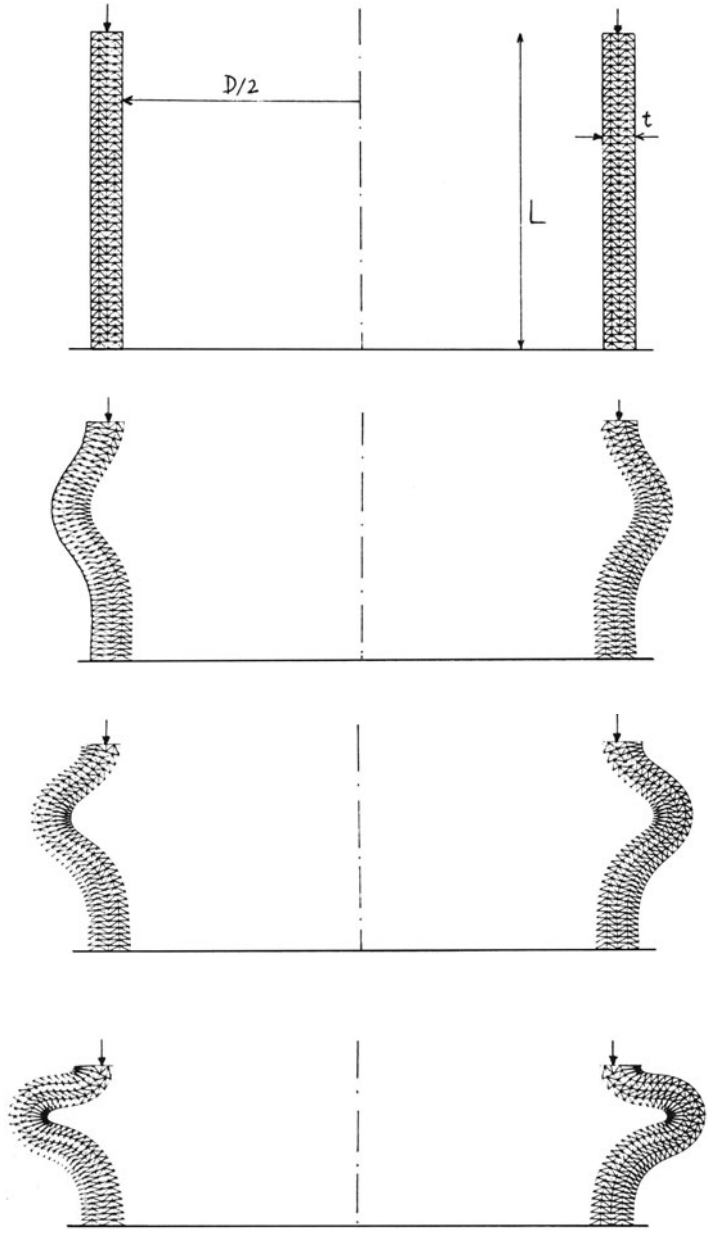


Figure 5 Axial Collapse of Perfectly Plastic Tube.

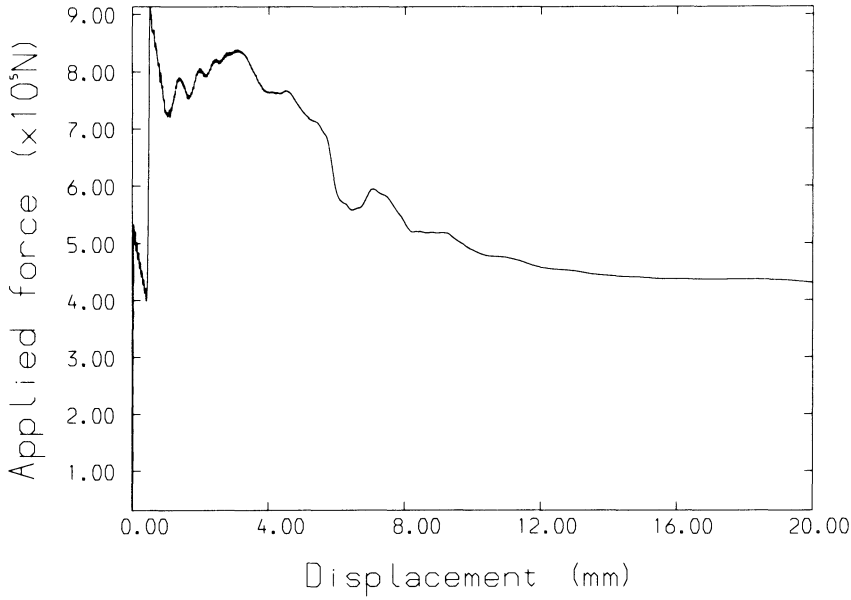


Figure 6 Load-deflection Curve for Perfectly Plastic Tube.

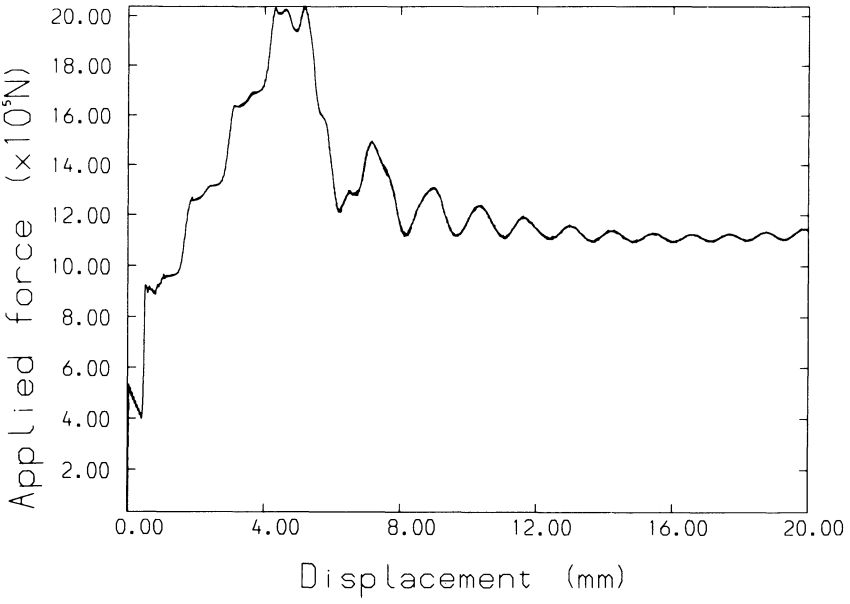


Figure 7 Load-deflection Curve for Strain-hardening Tube.

BOUNDARY ELEMENTS

APPLYING ANALYTICAL FUNCTIONS TO DEFINE SPECIAL BOUNDARY ELEMENTS IN A ELASTO-PLASTIC PROBLEM WITH SINGULARITIES IDEALIZED BY FINITE ELEMENTS

O. OHTMER

München

MESSERSCHMITT-BÖLKOW-BLOHM GMBH

Germany

SUMMARY

The named paper is based on the publications in [1], [2] and the literature named there. The procedure outlined in [1] is modified and then applied in a nonlinear analysis due to material nonlinearities. Based on the hybrid functionals in [2] nonlinear systems of equations in matrix formulation are derived. A modified NEWTON-RAPHSON-Procedure is applied as solution technique. In each iteration step the elements of a Finite Element Idealization are analysed due to a plastic or elastic deformation.
In those areas with singularities, where plastic deformations are expected so called singular boundary elements are used as Finite Element idealization. For these elements the stress- and displacement-distribution is expressed by analytical functions. Therefore the area integrals in the Hybrid-Stress Functional containing the parameters of variation in quadratic form are zero. The first and second variation are applied only to boundary integrals to generate the stiffness matrix. The original area is transformed to a circle or circular ring and then checked due to plastic deformations. If small elements $r \Delta \varphi \Delta r$ within the singular boundary element have a stress level that plastic deformations are occuring due to the Initial-Stress Approach equivalent fictive forces will be applied at the corners of the small elements.
Due to these fictive forces the elastic fictive deformation is equal to the real plastic deformation if the named iteration procedure has converged. The fictive forces are included in the assumed stress- and displacement-distribution as usual in the mathematical theory of elasticity.

The remaining domains are idealized by special triangular and quadrilateral boundary elements with straight edges. They are defined in the same manner as the singular boundary elements with the difference that now the triangular or quadrilateral area is transformed to a unit circle by special conformal mappings. Within these elements the equilibrium and compatibility equations are also fulfilled continuously. Therefore it is no longer necessary to idealize domains by a relative fine grid. As a consequence the problem can be described by a system of equations of relative small size.

Numerical Solution

Nonlinear Green's strain tensor:

$$\epsilon_{ij} = \frac{1}{2}\left(\underbrace{\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}} + \underbrace{\frac{\partial u_k}{\partial x_i} \cdot \frac{\partial u_k}{\partial x_j}}\right) ; i, j, k = 1, 2, 3$$

$$\epsilon = \{\epsilon_L\} + \{\epsilon_{NL}\} \qquad (1)$$

Approach to Finite Element Analysis:

The displacement fields u_i are expressed by the element joint displacements δ_i

$u = [N]\{\delta\}$; [N] is the matrix of "shape functions"

$\epsilon_L = [B_L] \cdot \{\delta\}$

$\epsilon_{NL} = [B_{NL}(\{\delta\})] \cdot \{\delta\}$

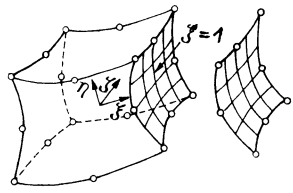


Fig. 1

Two-dimensional Problem:

$$\{u\} = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix}; \quad \{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{Bmatrix} \quad i \{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix}$$

displacements strains stresses

$v = u_y$

$$\{T\} = \begin{Bmatrix} T_x \\ T_y \end{Bmatrix} = \begin{Bmatrix} \sigma_x \cdot \eta_x + \sigma_{xy} \cdot \eta_y \\ \sigma_{xy} \cdot \eta_x + \sigma_y \cdot \eta_y \end{Bmatrix}$$

boundary tractions (2)

η_x, η_y direction cosinus of outward normal

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ & C_{22} & C_{23} \\ \text{sym.} & & C_{33} \end{bmatrix} = [S]^{-1}$$

stress-strain matrix

Hybrid functional (approach suggested by Pin Tong, T.H.H. Pian)

$$\pi_h = \sum_m \pi_{hm} \; ; \; \text{m elements}$$

$$\pi_{hm} = \int_{\partial A_m} \underbrace{(\tilde{u}_i - u_i) \cdot \sigma_{ij} \cdot \eta_j}_{(\tilde{u}_i - u_i) \cdot T_i} \cdot ds - \int_{(S\sigma)_m} \underbrace{\tilde{u}_i \cdot \bar{\sigma}_{ij} \cdot \eta_j}_{\tilde{u}_i \cdot \bar{T}_i} \cdot ds + \frac{1}{2} \int_{A_m} (2\,\sigma_{ij} \cdot \epsilon_{ij} - S_{ijkl} \cdot \sigma_{ij} \cdot \sigma_{kl})\, dA \qquad (3)$$

A_m area of m-th element; ∂A_m boundary of A_m; $\bar{T}_i$ prescribed boundary traction over the boundary $(S\sigma)_m$

$\left\{\tilde{u}\right\} = \begin{Bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{Bmatrix}$ displacement vector only defined over the element boundaries

S_{ijkl} compliance coefficients;

u_i; σ_{ij} displacements, stresses defined in A_m

$\tilde{u}_i$; T_i displacements, tractions only defined over ∂A_m

Euler equations of the functional (applying the principle of variation)

$$\begin{aligned}
\epsilon_{ij} &= S_{ijkl} \cdot \sigma_{kl} && \text{(condition of compatibility)} \\
\sigma_{ij,\,j} &= 0 && \text{(condition of equilibrium)} \\
T_i &= \sigma_{ij} \cdot \eta_j && (4) \\
u_i &= \tilde{u}_i \;\; \text{on} \;\; \partial A_m \\
T_i &= \bar{T}_i \;\; \text{on} \;\; (S\,\sigma)_m
\end{aligned}$$

Complex variable formulation of the stresses and displacements fulfil the conditions in A_m.

Simplified hybrid functional:

$$\pi_{hm} = \int_{\partial A_m} \{T\}^T \cdot \{\tilde{u}\} \cdot ds - \int_{\partial A_m} \{T\}^T \{u\} \cdot ds - \int_{(S\sigma)_m} \{\bar{T}\}^T \cdot \{\tilde{u}\}\, ds \qquad (5)$$

Proposed strategy:

Idealisation by singular substructures (superelements) around singularities (sharp corner, crack, concentrated force), using standard hybrid elements adjacent to the singular substructures or special triangular and quadrilateral 2D boundary elements.

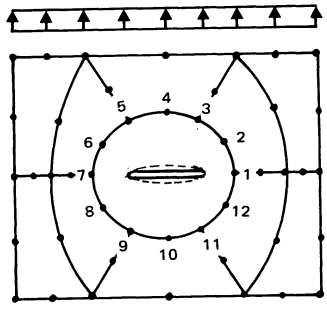


Fig. 2

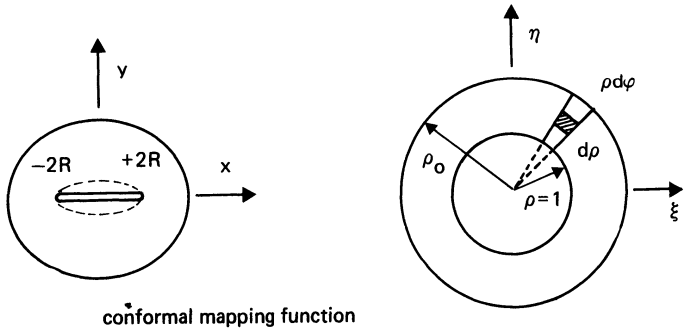


Fig. 3

conformal mapping function

$$Z = \omega(\zeta) = R\left(\zeta + \frac{1}{\zeta}\right);\quad R > 0 \tag{6}$$

Stress functions $\varphi(\zeta)$ and $\psi(\zeta)$ for a finite plane:

$$\varphi(\zeta) = -\frac{1}{2\pi(1+\kappa)} \sum_{k=1}^{M} (X_k + iY_k) \log(\omega(\zeta) - \omega(\zeta_k)) + \varphi_o(\zeta) \tag{7}$$

$$\psi(\zeta) = \frac{\kappa}{2\pi(1+\kappa)} \sum_{k=1}^{M} (X_k - iY_k) \log(\omega(\zeta) - \omega(\zeta_k)) + \psi_o(\zeta)$$

The vectors (X_k, Y_k) are perscribed concentrated forces (moments) at the points ζ_k. Given forces F or stresses with a continuous distribution on the boundary may be defined by a complex Fourier series. Therefore an analog expansion is assumed for $\varphi_o(\zeta)$ and $\psi_o(\zeta)$.

$$\varphi_o(\zeta) = \sum_{k=1}^{n} a_k \cdot \zeta^k; \quad \psi_o(\zeta) = \sum_{k=1}^{n} a_k^* \cdot \zeta^k; \quad F_1 + iF_2 = F = \sum_{k=-n}^{+n} A_k \cdot \theta^k$$

with

$\kappa = 3 - 4\nu$ for plane strain

$\kappa = (3 - \nu) / (1 + \nu)$ for plane stress

$z = \omega(\zeta)$ is the mapping function.

The displacement-field u + iv is defined by the two stress functions $\varphi(\zeta)$ and $\psi(\zeta)$.

$$\frac{E}{1+\nu} \cdot (u + iv) = \kappa \cdot \varphi(\zeta) - \frac{\omega(\zeta)}{\overline{\omega'(\zeta)}} \cdot \overline{\varphi'(\zeta)} \cdot - \overline{\psi(\zeta)} \tag{8}$$

The stresses σ_x, σ_y and σ_{xy} are defined by the stress functions $\varphi(\zeta)$, $\psi(\zeta)$ and the conformal mapping $\omega(\zeta)$.

$$\sigma_x + \sigma_y = 4\,\mathrm{Re}\left[\frac{\varphi'(\zeta)}{\omega'(\zeta)}\right]$$
$$\sigma_y - \sigma_x + 2i\sigma_{xy} = 2\,\frac{\overline{\omega(\zeta)}}{\omega'(\zeta)}\left[\varphi''(\zeta) - \frac{\omega''(\zeta)}{\omega'(\zeta)}\,\varphi'(\zeta)\right] + 2\,\frac{\psi'(\zeta)}{\omega'(\zeta)} \tag{9}$$

With the equations above one can express:

$$\{T\} = [R] \cdot \{\beta\} + [R_F] \cdot \{\beta_F\}\ ; \quad \{u\} = [U] \cdot \{\beta\} + [U_F]\{\beta_F\} \tag{10}$$

β vector of complex coefficients for $\{T\} = 0$ *)

$[R_F] \cdot \beta_F$ particular solution due to $\{\overline{T}\} \neq 0$

With

[L] the interpolation function matrix only defined on ∂A_m $\left\{\tilde{u}\right\}$ can be expressed:

$$\left\{\tilde{u}\right\} = \begin{bmatrix} 1 - \frac{S}{l} & o & \frac{S}{l} & o \\ o & 1 - \frac{S}{l} & o & \frac{S}{l} \end{bmatrix} \begin{Bmatrix} (u_x)_p \\ (u_y)_p \\ (u_x)_{p+1} \\ (u_y)_{p+1} \end{Bmatrix} = [L] \cdot \{q\} \tag{11}$$

l is the distance between the nodes p and p+1, $((u_x)_p, (u_y)_p)$ are the values of $\left\{\tilde{u}\right\}$ at the node p and s the distance on the boundary measured from p.

*) The generalized complex functions $\zeta_g^j = (\sum_{i=1}^{n} A_i \cdot \Theta_i(\xi,\eta) - 1) \cdot \zeta^j$ fulfil the boundary condition $\{T\} = 0$

$\Theta_j(\xi,\eta)$ are complex orthonormalized functions on the boundary.

Substitution of these equations in the simplified functional:

$$\pi_{hm} = \{\beta\}^T [G] \{q\} - \{\beta\}^T [H] \{\beta\} - \{\beta_F\}^T [G_F] \{q\}$$

with (12)

$$[G] = \int_{\partial A_m} [R]^T [L] \cdot ds; \quad [H] = \frac{1}{2} \int_{\partial A_m} ([R]^T [U] + [U]^T [R])\, ds$$

$$[G_F] = \int_{(S\sigma)_m} [R_F]^T [L] \cdot ds \; ; \quad [H_F] = \int_{Am} [R_F]^T \cdot [U]\, ds$$

$$\{J\} = \int_{(S\sigma)m} \{\bar{T}\}^T \cdot [L]\, ds$$

Derivation of π_{hm} with respect to $\{\beta\}^T$:
(first solution condition)

$$[G] \{q\} - [H] \{\beta\} = 0; \quad \{\beta\} = [H]^{-1} \cdot [G] \cdot \{q\}$$

Eliminating $\{\beta\}$ and derivating π_{hm} with respect to $\{q\}$ (second solution cond. $\{u\} = \{\tilde{u}\}$ on ∂A_m in the mean)

$$[k] \cdot \{q\} = \{p\} \tag{13}$$

with

$[k] = [G]^T [H]^{-1} [G]$ stiffness matrix of the singular substructure

$\{P\} = 2 [G]^T [H]^{-1} [H_F] \{\beta_F\} - [G_F]^T \cdot \{\beta_F\} + \{J\}^T$ element load vector

Updated LAGRANGE' Formulation

NEWTON-RAPHSON-ITERATION in each time step if necessary (see appendix I)

Numerical Time Integration by the WILSON- or NEWMARK-Method (see appendix II)

$$({}_{o}^{t}[K_L] + {}_{o}^{t}[K_{NL}]) \{\Delta u\}^{(i)} = {}_{o}^{t+\Delta t}\{P\} - {}_{o}^{t+\Delta t}\{\Delta F\}^{(i-1)} - [M] \{\ddot{u}\}^{(i)} \tag{14}$$

global stiffness matrix $\qquad i = 1, 2, 3, \ldots$

with $\quad {}^{t+\Delta t}\{u\}^{(i)} = {}^{t+\Delta t}\{u\}^{(i-1)} + \{\Delta u\}^{(i)}$

Application of the INITIAL STRESS Method

The INITIAL STRESS Method proposed by O. C. Zienkiewicz [2] to solve the elastic-plastic problem is applied, if no geometric nonlinearities are occuring. Due to plastic deformations the "Initial Stresses" are computed and applied in each iteration step. By the "Initial Stresses" the elastic structure is deformed as in the case of yielding due to a specific law.

The advantage is that no updating of the global stiffness matrix is necessary.

The check for plastic deformations and the computation of the "INITIAL STRESSES" due to the VON MISES- or DRUCKER-PRAGER yield condition is performed in each iteration step (see Appendix III).

. The original area is transformed to a circle or circular ring and then checked due to plastic deformations. If small elements $r \Delta\varphi \Delta r$ within the singular boundary element have a stress level that plastic deformations are occuring due to the Initial-Stress Approach equivalent fictive forces will be applied at the corners of the small elements. Due to these fictive forces the elastic fictive deformation is equal to the real plastic deformation if the named iteration procedure has converged. The fictive forces are included in the assumed stress- and displacement-distribution as usual in the mathematical theory of elasticity.

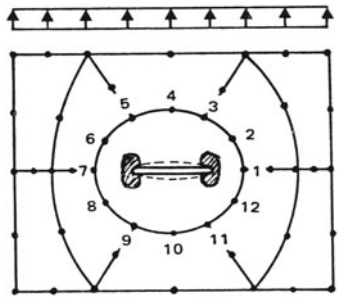


Fig. 4

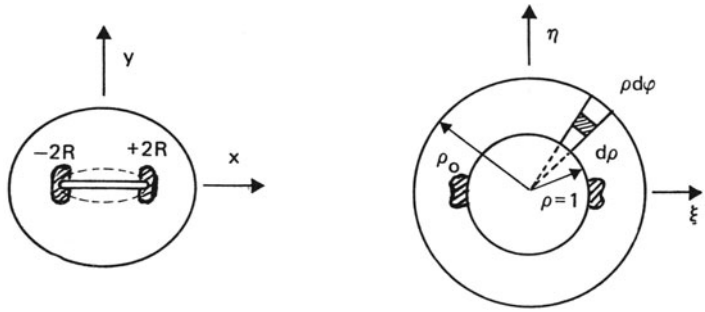


Fig. 5

At the crack tip the stress singularity $\sim 1/\sqrt{r}$ at time t = 0 is changing due to plastic deformation.

Special 2D Boundary Elements

These triangular and quadrilateral 2D Finite Elements fulfil the equilibrium- and compatibility equations continuously.

For the standard Finite Elements defined by displacement functions the equilibrium equations are only fulfilled at the joints not within the elements. In contradiction within the elements defined by analytical functions the equilibrium- and compatibility equations are fulfilled continuously except at the corners. Using these elements it is no longer necessary to idealize domains by a relative fine grid.

The conformal mapping of a triangle or a quadrilateral onto the unit circle is performed as outlined in [5]. The procedure is demonstrated for the regular triangle in Fig. 6. By (15) which holds also for a quadrilateral ($Z_0, Z_1 \ldots Z_n$ are the corner joints and $\pi a_0, \pi a_1 \ldots \pi a_n$ the associated angles) the triangle in Fig. 6 is transformed onto the smooth curvature a.
In a second step the curvature a is transformed onto the unit circle by orthogonal functions. Due to the definitions in [14] the complex functions $h_i(Z) = Z^{i-1}$, i = 1, 2, 3, ... are used to compute the scalar products F_{ij} in (16) for the domain with the curvature a. With given values F_{ij} in the Gram-Schmidt orthogonalization process the orthonormal functions $\Phi_i^*(Z)$ in (17) are computed. With (17) the conformal mapping $\omega = f_{(z)}$, $f_{(0)} = 0$ of the curvature a onto the unit circle is defined in (18).

$$\overset{n}{z}(z) = \left(\left((z - z_0)^{1/x_0} - \frac{1}{z_1}\right)^{1/x_1} - \frac{11}{z_2}\right)^{1/x_2} \ldots\ldots - \frac{n-11}{z_{n-2}}\Bigg)^{1/x_{n-2}} - \overset{n-1}{z}(0)\,. \tag{15}$$

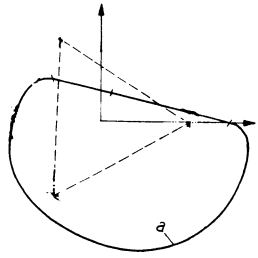


Fig. 6

$$F_{ij} = (h_i, h_j) = \int\int h_i(z)\,\bar{h}_j(\bar{z})\,dx\,dy\,, \tag{16}$$

$$\Phi_i^*(z) = \sum_{\nu=1}^{i} d_{i\nu}\,h_\nu(z) \tag{17}$$

$$f(z) = \sum_{\nu=1} b_\nu\,z^\nu \quad \text{with} \quad b_\nu = \sum_{i=\nu} d_{i1}\,\frac{d_{i\nu}}{\nu}\,. \tag{18}$$

Stress-Functions defined in a unit circle

$$\Phi_0(\xi) = \sum_{-n}^{+n} a_k \cdot \xi^k ; \qquad \psi_0(\xi) = \sum_{-n}^{+n} a_k^* \cdot \xi^k$$

$$\zeta = \xi + i \cdot \eta ; \quad a_k = \gamma_k + i\,\delta_k ; \quad a_k^* = \gamma_k^* + i\,\delta_k^*$$

$$\{a\} = \begin{Bmatrix} a_{-n} \\ \cdot \\ \cdot \\ \cdot \\ a_{-1} \\ 0 \\ a_{+1} \\ \cdot \\ \cdot \\ \cdot \\ a_{+n} \end{Bmatrix} ; \quad \{\gamma\} = \begin{Bmatrix} \gamma_{-n} \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{-1} \\ 0 \\ \gamma_{+1} \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{+n} \end{Bmatrix} ; \quad \{\delta\} = \begin{Bmatrix} \delta_{-n} \\ \cdot \\ \cdot \\ \cdot \\ \delta_{-1} \\ 0 \\ \delta_{+1} \\ \cdot \\ \cdot \\ \cdot \\ \delta_{+n} \end{Bmatrix} ; \quad \{\zeta\} = \begin{Bmatrix} \zeta^{-n} \\ \cdot \\ \cdot \\ \zeta^{-1} \\ 0 \\ \zeta^{+1} \\ \cdot \\ \cdot \\ \zeta^{+n} \end{Bmatrix} \tag{19}$$

$$\{a^*\} = \ldots ; \quad \{\gamma^*\} = \ldots \quad ; \quad \{\delta^*\} = \ldots\ldots \quad ; \quad \{\zeta\} = \ldots\ldots$$

$$\{\beta\} = \begin{Bmatrix} \{\gamma\} \\ \{\delta\} \end{Bmatrix} ; \quad \{\beta\}^* = \ldots ; \qquad [F] = \begin{bmatrix} 1 & & & & 0 \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & & 1 \end{bmatrix} \tag{20}$$

$$\Phi_0(\zeta) = \{\{\gamma\}^T + i\{\delta\}^T\}\{\zeta\}$$

$$\Phi_0(\zeta) = \{\{\zeta\}^T \,\vdots\, i\cdot\{\zeta\}^T\}\{\beta\} \quad ; \quad [N] = \begin{bmatrix} -n & & & & & & 0 \\ & \ddots & & & & & \\ & & -1 & & & & \\ & & & 0 & & & \\ & & & & +1 & & \\ & & & & & \ddots & \\ 0 & & & & & & +n \end{bmatrix}$$

$$\psi_0(\zeta) = \{\{\zeta\}^T \,\vdots\, i\cdot\{\zeta\}^T\}\{\beta\}^*$$

(21)

$$\Phi_0'(\zeta) = \left\{ ([N]\cdot\{\zeta\}\cdot\frac{1}{\zeta})^T \,\vdots\, i\,([N]\{\zeta\}\cdot\frac{1}{\zeta})^T \right\}\{\beta\}$$

$$\psi_0'(\zeta) = \left\{ ([N]\cdot\{\zeta\}\cdot\frac{1}{\zeta})^T \,\vdots\, i\,([N]\cdot\{\zeta\}\cdot\frac{1}{\zeta})^T \right\}\{\beta\}^*$$

$$\Phi_0''(\zeta) = \left\{ ([[N]-[F]]\cdot\{\zeta\}\cdot\frac{1}{\zeta^2})^T \,\vdots\, ([[N]-[F]]\cdot\{\zeta\}\frac{1}{\zeta^2})^T\cdot i \right\}\{\beta\}$$

Displacement-field:

$$\left\{ u \right\} = \left\{ \begin{matrix} u_x \\ u_y \end{matrix} \right\} = [U] \cdot \left\{ \beta \right\} + [U]^* \cdot \left\{ \beta \right\}^* + [U_F] \cdot \{\beta_F\}$$

with

$$\left\{ U_1 \right\} = \frac{3-\nu}{E} \cdot \left\{ \zeta \right\} - \frac{1+\nu}{E} \cdot \frac{\omega(\xi)}{\omega'(\xi)} \cdot \left([N] \cdot \left\{ \bar{\xi} \right\} \cdot \frac{1}{\bar{\zeta}} \right) \tag{22}$$

$$[U] = \left[\begin{array}{c|c} \mathrm{Re} \left\{ U_1 \right\}^T & \mathrm{Re}\left(i \left\{ U_1 \right\}^T\right) \\ \hline \mathrm{Im} \left\{ U_1 \right\}^T & \mathrm{Im}\left(i \left\{ U_1 \right\}^T\right) \end{array} \right] ; \quad [U]^* = \left[\begin{array}{c|c} \mathrm{Re} \left\{ \bar{\xi} \right\}^T & \mathrm{Re}\left(i \left\{ \bar{\xi} \right\}^T\right) \\ \hline \mathrm{Im} \left\{ \bar{\xi} \right\}^T & \mathrm{Im}\left(i \left\{ \bar{\xi} \right\}^T\right) \end{array} \right]$$

Boundary traction-field:

$$\left\{ T \right\} = \left\{ \begin{matrix} T_x \\ T_y \end{matrix} \right\} = \left\{ \begin{matrix} n_x \cdot \sigma_x + n_y \cdot \sigma_{xy} \\ n_x \cdot \sigma_{xy} + n_y \cdot \sigma_y \end{matrix} \right\} = [R] \cdot \left\{ \beta \right\} + [R]^* \cdot \left\{ \beta \right\}^* + [R_F] \left\{ \beta_F \right\}$$

with (23)

$$\left\{ R_1 \right\} = \frac{1}{\zeta \cdot \omega'(\xi)} \cdot [N] \cdot \left\{ \zeta \right\} ; \quad \left\{ R_2 \right\} = \frac{1}{\zeta^2} \cdot [[N] - [F]] \left\{ \zeta \right\}$$

$$\left\{ R_3 \right\} = \frac{\omega(\xi)}{\omega'(\xi)} \cdot \left(\left\{ R_2 \right\} - \omega''(\xi) \cdot \left\{ R_1 \right\} \right)$$

$$[R] = \left[\begin{array}{c|c} \mathrm{Re}\left(2 \left\{ R_1 \right\}^T - \left\{ R_3 \right\}^T\right) \cos\varphi & \mathrm{Im}\left(2 \left\{ R_1 \right\}^T - \left\{ R_3 \right\}^T\right) \cos\varphi \\ + & + \\ \mathrm{Im}\left(\left\{ R_3 \right\}^T\right) \cdot \sin\varphi & \mathrm{Re}\left(\left\{ R_3 \right\}^T\right) \sin\varphi \\ \hline \mathrm{Re}\left(2 \left\{ R_1 \right\}^T + \left\{ R_3 \right\}^T\right) \sin\varphi & \mathrm{Im}\left(2 \left\{ R_1 \right\}^T + \left\{ R_3 \right\}^T\right) \sin\varphi \\ + & + \\ \mathrm{Im}\left(\left\{ R_3 \right\}^T\right) \cos\varphi & \mathrm{Re}\left(\left\{ R_3 \right\}^T\right) \cos\varphi \end{array} \right]$$

$$[R]^{*} = \begin{bmatrix} -\mathrm{Re}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\cos\varphi & -\mathrm{Im}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\cos\varphi \\ \mathrm{Im}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\sin\varphi & \mathrm{Re}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\sin\varphi \\ \hline \mathrm{Re}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\sin\varphi & \mathrm{Im}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\sin\varphi \\ \mathrm{Im}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\cos\varphi & \mathrm{Re}\left(\left\{R_1\right\}^{+}\left\{T\right\}\right)\cos\varphi \end{bmatrix}$$

$$\underset{n_x}{\sin\alpha} + i\underset{n_y}{\cos\varphi} = e^{i\alpha} = e^{i\varphi}\cdot\frac{\omega'(\xi)}{|\omega'(\xi)|} \qquad (24)$$

Practical Idealization of a structure with singularities

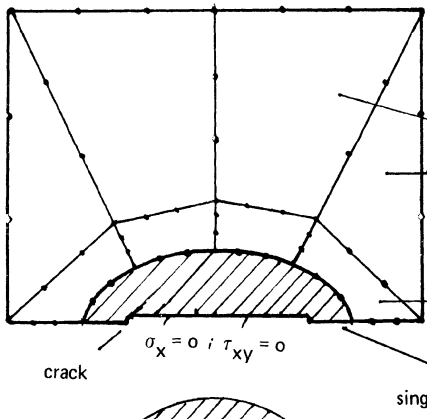
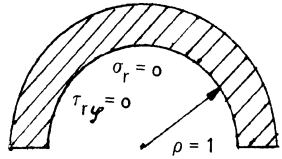


Fig. 7



$\sigma_r = o; \qquad \tau_{ry} = o :$

Fig. 8

$$a_k^{*} = -(k+1)\cdot a_{k+2} + \bar{a}_{-k-2} \qquad (25)$$

$k = \ldots . -4, -3, -1, 0, 1, 2, 3, \ldots$

The boundary conditions $\delta_r = o$; $\tau_{r\varphi} = o$ in the singular boundary element lead to coupled stress functions in (25).

$$\left\{\beta\right\}^{*} = f_1\left(\left\{\beta\right\}\right); \quad \psi(\zeta) = f_2(\Phi(\zeta)) \qquad (26)$$

Interpolation of the displacements on the boundary:

$$\left\{\tilde{u}\right\} = [L] \cdot \left\{q\right\}$$

There is no need to assume u over the crack surface because the tractions are zero.

with

$$\left\{q\right\} = \begin{Bmatrix} u_p \\ v_p \\ u_{p+1} \\ v_{p+1} \\ u_{p+2} \\ v_{p+2} \end{Bmatrix} ; \qquad (27)$$

$u = u_x; \quad v = v_y$

$\tilde{u}$ varying as a quadratic function between nodes p and p+2; (p = 1, 3, 5, ...)

$$[L] = \begin{bmatrix} (1-\frac{s}{l})(1-\frac{2s}{l}) & 0 & 4\frac{s}{l}(1-\frac{s}{l}) & 0 \\ 0 & (1-\frac{s}{l})(1-\frac{2s}{l}) & 0 & 4\frac{s}{l}(1-\frac{s}{l}) \end{bmatrix} \qquad (28)$$

s = length parameter; l = length between two joints.

$$\begin{bmatrix} \frac{s}{l}(2\frac{s}{l}-1) & 0 \\ 0 & \frac{s}{l}(2\frac{s}{l}-1) \end{bmatrix}$$

Formulation of the Hybrid Element Stiffness Matrix [K]

$$[G] = \int_{\text{unit circle}} [[R]^* + [R]^*]^T \cdot [L] \cdot \underbrace{((\frac{d\,\mathrm{Re}\,\omega(\varphi)}{d\varphi})^2 + (\frac{d\,\mathrm{Im}\,\omega(\varphi)}{d\varphi})^2)^{1/2} \cdot d\varphi}_{ds}$$

$$[H] = \int_{\text{unit circle}} [[R] + [R]^*] \cdot^T [[U] + [U]^*] + [[U] + [U]^*]^T [[R] + [R]^*] \; ds \qquad (29)$$

Applying the Trapez-Integration-Rule due to periodic functions.

$$[K] = \frac{1}{2} [G]^T \cdot [H]^{-1} [G] \qquad \text{Substructure (Superelement) Stiffness Matrix} \qquad (30)$$

References

[1] O. Ohtmer:
Applying Analytical Functions in a Boundary Value Problem with Sinularities idealized by Finite Elements. Paper presented at the Second World Congress on Finite Element Methods, Bournemouth 1978, England

[2] R.L. Spilker and T.H.H. Pian
Hybrid-Stress Models for Elastic-Plastic Analysis by the Initial-Stress Approach
International Journal for Numerical Methods in Engineering Vol. 14.359-378 (1979)

[3] Barsoum R. S.
On the use of isoparametric Finite Elements in linear fracture mechanics. Int. Journal Num. Methods in Engineering Vol. 10, 25-37 (1976).

[4] Ohtmer, O.
Konforme Abbildung einfach zusammenhängender Gebiete mit Ecken und starken Krümmungen der Randkurven und ihre Anwendung bei der Lösung von harmonischen und biharmonischen Randwertproblemen.
Dissertation, T. U. Braunschweig 1968.

[5] Ohtmer, O.
Zur angenäherten konformen Abbildung von Gebieten mit Ecken und starken Krümmungen der Randkurven auf den Einheitskreis, ZAMM 48, GAMM-Tagung 1968.

[6] Ohtmer, O.
Berechnung von zwei- und dreidimensionalen Strömungsproblemen nach der Substructure-Theorie mit isoparametrischen finiten Elementen ZAMM 54, GAMM-Tagung 1973.

[7] Pin Tong, T. H. H. Pian and S. J. Lasry
A Hybrid-Element Approach to Crack Problems in Plane Elasticity. Int. Journal Num. Methods in Engineering, Vol. 7, 297-308 (1973).

[8] Muskhelishvili
Some Basic Problems of the Mathematical Theory of Elasticity
P. Noordhoff, Groningen 1963 The Netherlands

[9] ICES-STRUDL, Operation-Application-Manuals, MBB 1976

[10] T. H. H. Pian and Pin Tong
Basis of Finite Element Methods for Solid Continua
Int. Journal Num. Meth. in Eng. Vol. 1, 3-28 (1969)

[11] Pin Tong
A Hybrid Crack Element for Rectilinear Anisotropic Material
Int. Journal Num. Meth. in Eng. Vol. 11, 377-403 (1977).

[12] K. Washizu
Variational Methods in Elasticity and Plasticity
Pergamon Press, London 1968

[13] P. Tong
New Displacement Hybrid Finite Element Method for solid Continua.
Int. Journal Num. Methods Eng. Vol. 2, 73-83 (1970)

[14] St. Bergmann and S.G. Herriot
Application of the method of the kernel function for solving boundary value problems, Num. Mathematik 3, p. 209-225 (1961)

Appendix I

NEWTON-RAPHSON-Iteration-Method to solve nonlinear equations

$$f_1(x_1, x_2 \ldots) = a_{11} + b_{11}\,x_1 + c_{11}\,x_1^2 + a_{12} + b_{12}\,x_2 + c_{12}\,x_2^2 + d_{12}\,x_1 \cdot x_2 \ldots = 0$$

$$f_2(x_1, x_2 \ldots) = a_{21} + b_{21}\,x_1 + c_{21} \cdot x_1^2 + a_{22} + b_{22} \cdot x_2 + c_{22}\,x_2^2 + d_{22}\,x_1\,x_2 \ldots = 0$$

. . . . . .

$$[D] = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \ldots \ldots \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \ldots \ldots \\ \ldots \ldots \ldots \end{bmatrix}$$

$$\left[D\left(\{x_j\}\right)\right] \cdot \{z\} = \left\{f\left(\{x_j\}\right)\right\}$$

$$\{x_{j+1}\} = \{x_j\} - \{z\}$$

Appendix II

Numerical Time Integration in Dynamic Analysis
(Newmark-Beta-Method)

$$\overset{t+\Delta t}{\{\ddot{U}\}} = \frac{1}{a \cdot \Delta t^2} \cdot \left(\overset{t+\Delta t}{\{U\}} - \overset{t}{\{U\}}\right) - \frac{1}{a \cdot \Delta t} \overset{t}{\{\dot{U}\}} - \left(\frac{1}{2a} - 1\right) \cdot \overset{t}{\{\ddot{U}\}}$$

$$\overset{t+\Delta t}{\{\dot{U}\}} = \frac{\delta}{a \cdot \Delta t} \left(\overset{t+\Delta t}{\{U\}} - \overset{t}{\{U\}}\right) + \left(1 - \frac{\delta}{a}\right) \overset{t}{\{\dot{U}\}} + \left(1 - \frac{\delta}{2a}\right) \cdot \Delta t \cdot \overset{t}{\{\ddot{U}\}}$$

Δt time step

$$a \geq \frac{1}{4}\left(\delta + \frac{1}{2}\right)^2 ; \quad \delta \geq \frac{1}{2} \quad ; \quad a, \delta \text{ constants}$$

Appendix III

Stress-Strain Matrix in 2D Elastic-Plastic Analysis
Material isotropic hardening

VON MISES yield condition

$$i_2 = \frac{1}{6}\left((\sigma_{11}-\sigma_{22})^2 + (\sigma_{11})^2 + (\sigma_{22})^2\right) + (\sigma_{12})^2 = \frac{1}{3}(\sigma_F)^2$$

$\overset{t}{\sigma}_F$ yield stress at time t

$$[C]^{EP} = \frac{E}{1+\nu}\begin{bmatrix} & \epsilon_{11} & \epsilon_{22} & \epsilon_{12} \\ \epsilon_{11} & \frac{1-\nu}{1-2\nu} - \beta\cdot(\overset{t}{S}_{11})^2 & \frac{\nu}{1-2\nu} - \beta\cdot\overset{t}{S}_{11}\cdot\overset{t}{S}_{22} & -\beta\cdot\overset{t}{S}_{11}\cdot\overset{t}{S}_{12} \\ \epsilon_{22} & & \frac{1-\nu}{1-2\nu} - \beta\cdot(\overset{t}{S}_{22})^2 & -\beta\cdot\overset{t}{S}_{22}\cdot\overset{t}{S}_{12} \\ \epsilon_{12} & \text{symmetric} & & \frac{1}{2} - \beta\cdot(\overset{t}{S}_{12})^2 \end{bmatrix}$$

$$\beta = \frac{3}{2}\,\frac{1}{(\overset{t}{\sigma}_F)^2}\left(\frac{1}{1+\frac{2}{3}\cdot\frac{E\cdot E_T}{E-E_T}\cdot\frac{1+\nu}{E}}\right)$$

$\overset{t}{\sigma}_{ij}$ = total stresses at time t

$\overset{t}{\sigma}_m = \overset{t}{\sigma}_{ii/3}$ mean stresses at time t

$\overset{t}{S}_{ij} = \overset{t}{\sigma}_{ij} - \overset{t}{\sigma}_m\cdot\delta_{ij}$ deviatoric stresses

E = Young's module

E_T = Strain hardening module

ν = Poisson ratio

Examples

Example 1: The crack in Fig. 2 within a double connected region is transformed on to a circular ring by the mapping function:

$$z = \Omega(\zeta^*) = A\left(\zeta^* + \frac{1}{\zeta^*}\right), \quad A > 0.$$

Example 2: The sharp corner in the hole in Fig. 9 is deleted by the mapping function (15a) see [4], [5], [11].

$$z = \zeta^{\pi/a\pi}; \qquad (1 \leqq a \leqq 2), \tag{15a}$$

$$z'(\zeta) = \frac{1}{a}\,\zeta^{(1/a-1)} = \frac{1}{a} \cdot r^{(1/a-1)} \cdot \{\cos(1/a-1)\,\theta + i \cdot \sin(1/a-1)\,\theta\},$$

$z'(\zeta)$ the so called "Edge-Function" shows the strain – (velocity) singularities.

a = 2 (crack) $\rightarrow r^{-1/2}$,
a = 3/2 (corner, 90°) $\rightarrow r^{-1/3}$,
a = 1 (smooth boundary) $\rightarrow r^{0}$.

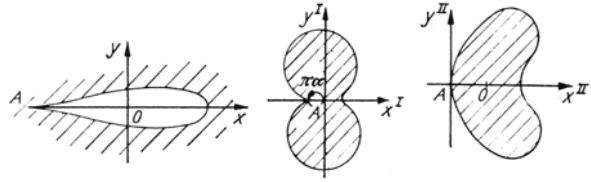


Fig. 9

Example 3: The crack with free surfaces in Fig. 10, a circle with a cut is transformed on to a circle by the function:

$$z(\zeta) = \frac{\omega_4 + \mathrm{tg}^2(a/8)}{1 + \mathrm{tg}^2(a/8) \cdot \omega_4}; \qquad \omega_4 = \omega_3^2 = \left(\frac{\omega_2 - 1}{\omega_2 + 1}\right)^2;$$

$$\omega_2 = \sqrt{\omega_1} = \left(\frac{e^{i(a/2)} - \zeta \cdot e^{-i(a/2)}}{1 - \zeta}\right)^{1/2}.$$

By $\mathrm{tg}^2(a/8)$ the length of the cut is specified.

Fig. 10

Example 4: A sector of a circle loaded at the corner by a concentrated force is mapped on to a circle in Fig. 11 .

$$z(\zeta^*) = \left(\frac{1-\sqrt{\zeta^*}}{1+\sqrt{\zeta^*}}\right)^{\alpha}$$

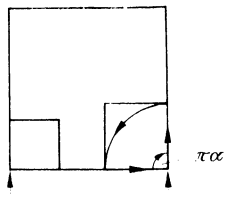


Fig. 11

Example 5

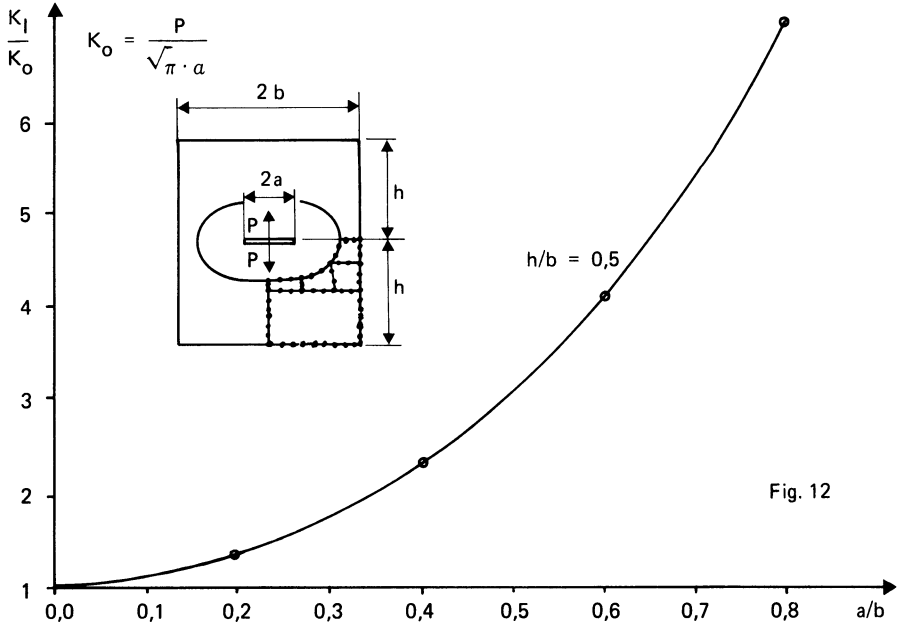


Fig. 12

Central crack in a rectangular sheet with opposing forces at the centre of the crack

BEASY A Boundary Element Analysis System

D.J. DANSON, C.A. BREBBIA & R.A. ADEY
CM Consultants, Ashurst Lodge,
Southampton, England

1. INTRODUCTION

Over the past decade the finite element method (FEM) has become established as a valuable tool in the solution of a wide variety of problems in engineering. The FEM may be seen as a method of solving boundary value problems where the phenomenon in the domain being studied obeys known differential equations. In the FEM the domain is discretized into a number of elements in each of which the solution of the governing equation is approximated by some function which satisfies the boundary conditions. A set of equations is then set up which when solved forces the solution at various points in the domain, known as nodal points, to the best approximation allowed by the approximating functions and the boundary conditions.

An alternative approach is to use functions which satisfy the differential equation in the domain but not the boundary conditions. The boundary may be divided into elements and the boundary values assumed to vary in some manner within these elements. A set of equations may then be formulated in terms of nodal values, with the nodes this time only on the boundaries. The solution of these equations forces the best solution permitted by the assumption of boundary value variation along each boundary element. The attractions of such an approach are obvious. Only the boundary need be discretized thus reducing by one dimension the long list of nodal coordinates and connectivity tables which make the finite element method so tedious. The resulting equations are much fewer in number and so may be solved more readily on computers with limited storage. This idea is the basis of the method known as boundary elements [1,2]. Boundary elements are also well adapted to problems with infinite boundaries.

The mathematical formulation of the boundary element method (BEM) is more complicated than in the FEM. Nevertheless the BEM has been applied successfully to problems in potential theory, elasticity, plasticity and time dependent problems such as those governed by the diffusion equation and the wave equation.

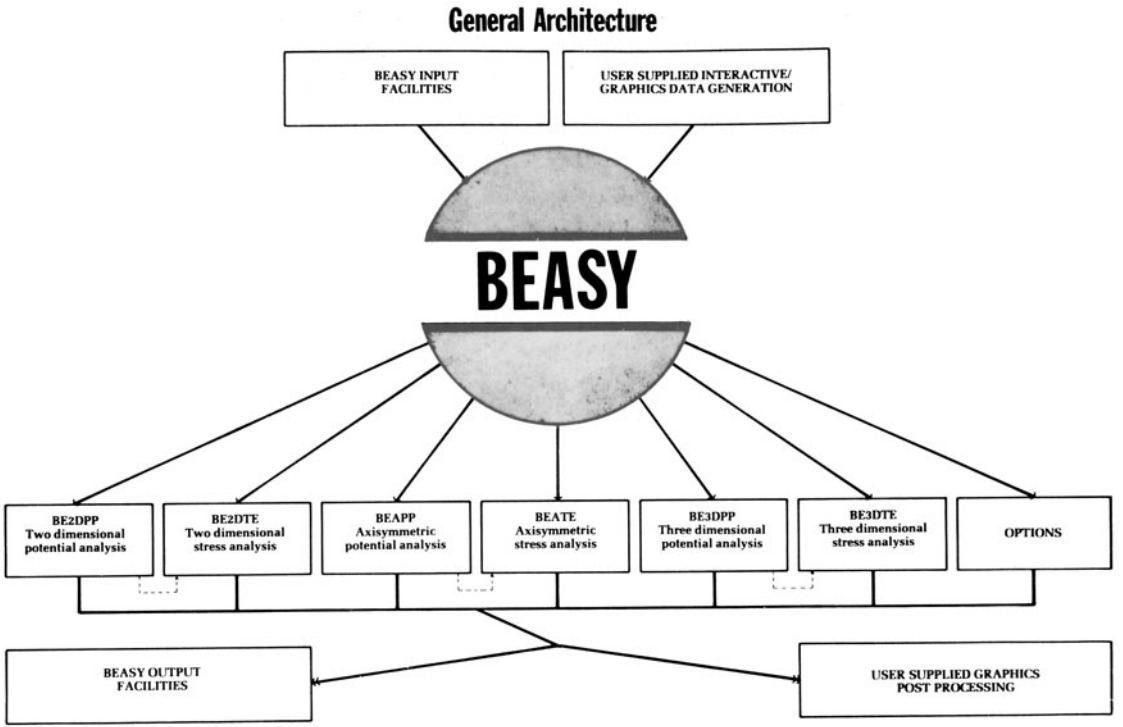


Figure 1

2. SPECIFICATION

The general architecture of BEASY is shown in Fig. 1.

As can be seen from Fig. 1 BEASY consists of six independent modules for the solution of problems in potential theory and linear isotropic stress analysis. The box labelled OPTIONS covers various more advanced applications of the BEM.

Element Types

BEASY uses constant, linear and quadratic non-conforming boundary elements. These are illustrated in Fig. 2.

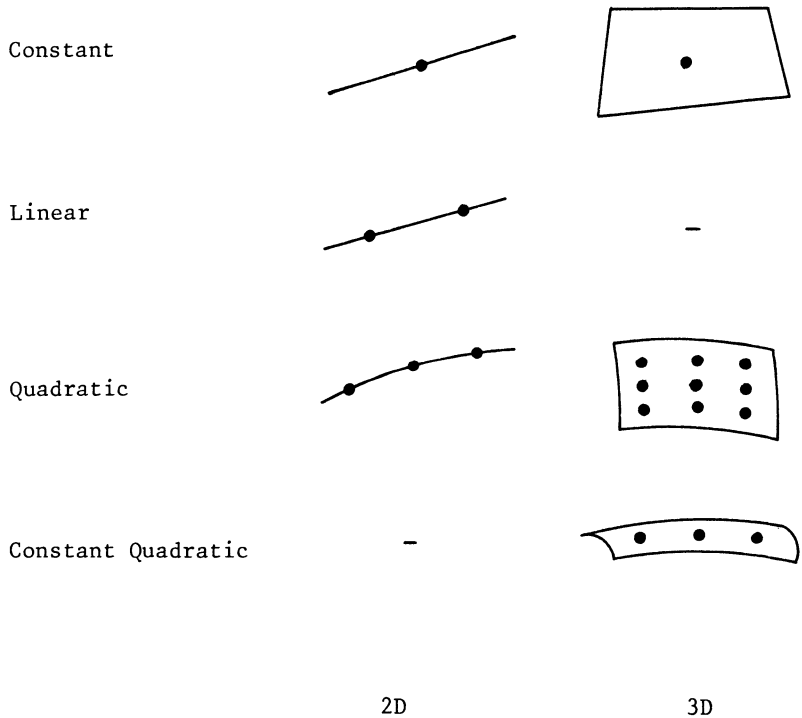


Figure 2 BEASY's Elements

The constant and linear elements have linear geometry described by linear shape functions and the quadratic and constant/quadratic elements have fully quadratic geometry described by quadratic shape functions. The elements are called non-conforming because the nodes are situated within the element boundaries rather than on the edge of the element. The reason for choosing this type of element is practical rather than theoretical. It is very easy to mix non-conforming elements putting quadratic elements in areas of rapidly

varying stresses (or potential) and linear or constant elements elsewhere. It is not necessary for the points defining the boundary element mesh, called "mesh points", and which would be nodes if conforming elements were used, to be common to several elements. Fig. 3 shows part of a mesh for a 3D problem which is a perfectly valid mesh for BEASY but would be difficult to handle if conforming elements were used. There are no problems of "fanning out" elements from areas of high element density.

At the moment the 3D modules do not contain a linear element since most of the time users are content to use either constant or quadratic elements. This linear element will be added, together with triangular elements, in the next release of BEASY.

The 3D quadratic element is a complete bi-quadratic element containing nine nodes as opposed to the more common incomplete 8-node quadrilateral. The reason for choosing a 9-node element is not only because it contains all the terms for a full bi-quadratic expansion but because the classical BEM is a collocation technique with the nodes as collocation points and the use of a 9-node element gives rise to an even pattern of collocation points. The necessity for this is borne out by the fact that the 9-node element behaves much better than the 8-node one particularly in problems where a change in the type of boundary condition produces a singularity in the flux (for potential problems) or in the stresses (in elasticity problems).

The constant/quadratic element which has three nodes and allows the system variables to vary quadratically in one direction but allows no variation in the other is surprisingly useful and has been used quite extensively on problems involving pipelines.

Applied Loads and Boundary Conditions

The boundary conditions allowed for in the potential modules are potential or flux density prescribed or a linear relation between the two (generally called heat transfer boundary condition as it is a very common boundary condition in thermal

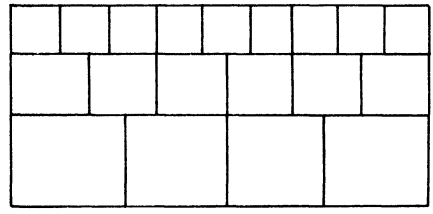


Figure 3 Part of a Typical BE Mesh

analyses). Non-linear boundary conditions as specified by clients for a particular application are easily modelled and this has been done successfully in the field of cathodic

protection where the potential and flux (current) density on the cathode are related by the polarization curve. The 2D and 3D modules can also model concentrated point and line sources which is a useful feature for many applications.

The stress analysis modules allow for prescribed displacement or loads or spring boundary conditions. The boundary conditions may be entered either in the global coordinate system or in a local system, one of whose axes always coincides with the normal to the boundary surface. This local system is not only useful when applying boundary conditions of a single type (e.g. when specifying an internal pressure in a spherical pressure vessel) but is absolutely essential when specifying a mixed type of boundary condition such as sliding, where the displacement is prescribed normal to the boundary and the loading is prescribed tangential to the boundary. Problems with gravitational or rotational loading and problems where the stresses are due to steady state thermal loading may also be analysed. These problems involving body forces may be analysed quite simply [3]. In the case of thermal analyses it is necessary to first solve the potential problem to obtain the temperatures and fluxes at the boundary nodes and the temperatures at any internal points. This information is then fed to the stress analysis module which then calculates the thermal stresses. Exactly the same data file may be used for both analyses.

Analysis Steps

BEASY carries out a typical Boundary Element analysis in six distinct steps. (see Table 1). The analysis may be started or stopped at each step.

BEASY carries out some fairly simple checks on the data. If clients request the addition of more sophisticated checks then these can be provided.

The boundary element method results from applying the techniques of finite element discretization to the boundary integral formulation of the problem. This results in the formation of influence matrices (usually H and G) which describe the behaviour at each node due to unit excitation at

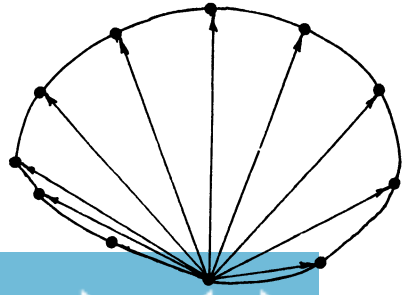


Figure 4 Influence Factors

Step	Input Required	Computation	Output Generated
1	Data file	Check on data	
2	Data file	Formation of influence matrices	Influence matrices
3	Data file Influence matrices	Application of the type of boundary conditions to form the system matrix	System matrix $\underset{\sim}{A}$ (Part of system of equations $\underset{\sim}{A}\ \underset{\sim}{x} = \underset{\sim}{d}$)
4	Data file System matrix	Reduction of the left hand side of the system of equations	Reduced left hand side
5	Data file Reduced left hand side	Application of the magnitude of the boundary conditions to form the right hand vector $\underset{\sim}{d}$ Reduction and back-substitution to obtain the boundary solution	Boundary solution
6	Data file Boundary solution	Computation of results at internal points	Results at internal points

Table 1

each node (Fig. 4). The influence matrices $\underset{\sim}{H}$ and $\underset{\sim}{G}$ are related by the equation

$$\underset{\sim}{H}\ \underset{\sim}{u} = \underset{\sim}{G}\ \underset{\sim}{p} + \underset{\sim}{b}$$

where $\underset{\sim}{u}$ is the vector of nodal boundary potentials or displacements

$\underset{\sim}{p}$ is the vector of nodal boundary fluxes or loadings

and $\underset{\sim}{b}$ is a vector which results from sources within the problem or from body forces.

The $\underset{\sim}{H}$ and $\underset{\sim}{G}$ matrices are analogous to the stiffness matrix which one obtains when using the Finite Element Method (FEM). The fact that there are two matrices and not one is a consequence of the mixed character of BEM solutions, i.e., one works with potential and fluxes, displacements and stresses, etc. This mixed approach makes BEM results generally more accurate than FEM solutions.

Application of the type of boundary conditions enables the

elements of the matrices to be rearranged so that we may write

$$\underset{\sim}{A}\ \underset{\sim}{x} = \underset{\sim}{B}\ \underset{\sim}{f} + \underset{\sim}{b}$$

where $\underset{\sim}{A}$ is the system matrix
$\underset{\sim}{x}$ is the vector of unknowns
$\tilde{B}$ is the complementary matrix
and $\tilde{\underset{\sim}{f}}$ is a vector of prescribed boundary values.

The system matrix $\underset{\sim}{A}$ and the complementary matrix $\underset{\sim}{B}$ are stored on disc.

Gauss elimination is now applied to the system matrix $\underset{\sim}{A}$ so that it may be expressed as the product of an upper and lower triangular matrix

$$\underset{\sim}{L}\ \underset{\sim}{U}\ \underset{\sim}{x} = \underset{\sim}{B}\ \underset{\sim}{f} + \underset{\sim}{b}$$

This reduction of the left hand side also consumes a large proportion of the total run time for a typical problem. The matrices $\underset{\sim}{L}$ and $\underset{\sim}{U}$ are stored on disc.

Not until this stage is the vector $\underset{\sim}{b}$ due to sources/body forces evaluated and the right hand side vector $\underset{\sim}{d}$ calculated from the equation

$$\underset{\sim}{d} = \underset{\sim}{B}\ \underset{\sim}{f} + \underset{\sim}{b}$$

We now have the equation

$$\underset{\sim}{L}\ \underset{\sim}{U}\ \underset{\sim}{x} = \underset{\sim}{d}$$

which may be solved for the vector of unknowns x by the normal right hand side reduction/backsubstitution process.

BEASY uses an out-of-core block solver based on the one published by Das [4]. This enables large problems to be solved efficiently on quite small machines.

The final step calculates the values at internal points using the boundary solution just obtained.

The reason for carrying out the analysis in the manner described above is to minimize the run times required for repeated analyses. The influence matrices are dependent only on the mesh geometry. Once the user has got that right he should never need to form the influence matrices more than once. The analysis for subsequent boundary conditions need only be restarted at step 3 if the type of boundary condition has been altered. e.g. a prescribed displacement is specified where previously the loading was specified. If only the magnitude of the boundary condition is altered, or new sources/

body forces added then it is not necessary to repeat steps 3 and 4 but the run may be restarted at step 5. Thus both the time consuming steps of forming the influence matrices and of decomposing the system matrix are avoided.

Once the boundary solution is obtained the solution at internal points is calculated by a fairly simple procedure. No further equation solution is required. If after looking at the results the user decides results are required at some more internal points then only step 6 needs to be repeated.

Symmetry

Symmetry is handled by the simple expedient of reflecting the boundary about the plane of symmetry and continuing the boundary integration around the reflected part of the structure. By making use of the fact that not only the geomtry but also the potentials,fluxes, displacements and loadings are symmetric the number of equations is not increased. Indeed it is reduced by the fact that no elements are required on the plane of symmetry. The time taken to compute the influence matrices is increased but the extra cost is nearly always outweighed by the reduced number of man-hours taken up in data preparation.

Zones or Subregions

Although BEASY does not enable the user to model problems with continuously varying material properties such as Young's modulus or conductivity piecewise constant properties may be handled by considering each part of the problem as a boundary element zone or subregion. This is a very useful facility as it may also be used in certain types of problems to reduce the total run time as its effect is to start to band the influence matrices.

BEASY Options

The options in BEASY cover a number of more advanced BEM applications. They each have only one element type and cannot handle zoned problems. They are:-

BETA2D Solves the diffusion equation in two dimensions. Its main application is in time-dependent thermal calculations. Uses a conforming linear element.

BETAAX As BETA2D but for axisymmetric geometries.

BEPLAS Solves problems in elasto-plasticity in two dimensions. Uses a conforming linear element.

BEPLAX As BEPLAS but for axisymmetric geometries.

BEREPOT Solves the scalar/potential wave equation in three dimensions. Has been used to analyse the transient response of liquids due to an explosion at a point [5]. Uses a non-conforming complete biquadratic quadrilateral element.

3. EXAMPLES

Fig. 5 shows the BE mesh used for stress analysis of a connecting rod when subjected to a load from the crankshaft. Notice that no elements are required on the plane of symmetry. All the necessary commands to create the mesh and loading are given below.

```
LE 3
SX
BP 1,0,0
BP 2,35,0
BP 3,35,47.803
BP 4,37,56.75
BP 5,55,95
BP 6,63.759,100.56
BP 7,67,100.56
BP 8,67,144.29
BP 9,65,754,155
BP 10,52.846,159.64
BP 11,0,187.5
BP 12,0,167.5
BP 13,0,82.5
BP 14,0,125
BP 15,56.1,47.803
BP 16,63.759,90.88
BP 17,65.754,175.7
BL 1,1,2,2
BL 2,2,3,4
BC 3,3,4,15,1
BL 4,4,5,5
BC 5,5,6,16,2
BL 6,6,7,1
BL 7,7,8,3
BL 8,8,9,2
BC 9,9,10,17,2
BC 10,10,11,14,6
BC 11,12,13,14,11
ZE 2100
ZP 0.28
ZI 1,4.135,66
ZI 9,37.218,66
ZI 10,42.723,144.29
ZI 14,32.145,144.29
EP 29,-1,-7.802,-7.705,-7.775
EP 30,-1,-7.705,-6.947,-7.424
EP 31,-1,-6.947,-5.533,-6.293
EP 32,-1,-5.533,-3.522,-4.586
EP 33,-1,-3.522,-1.231,-2.442
PD 1,-1,0,0
```

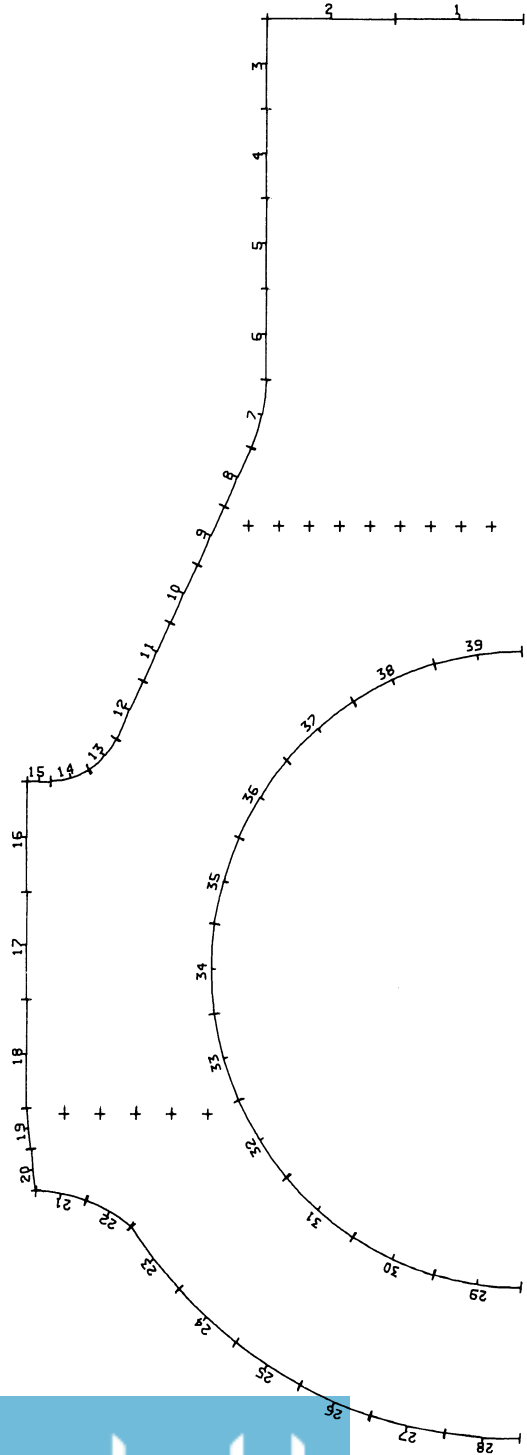


Figure 5 BE Mesh for a Connecting Rod
Lines of internal points marked by +'s have been created at sections at which the designer is particularly interested in the stresses. The example illustrates how easy it is to prepare a BE mesh.

Corrosion Protection System for an Offshore Platform

Fig. 6 shows the boundary element mesh required for the analysis of the impressed current corrosion protection system for an offshore platform to be operated in the North Sea. The requirement is to solve Laplace's equation in the infinite domain of the seawater and the designer wishes to know the current density on the cathode, which is the hull of the platform. Since the boundary of the seawater is the platform itself this example is ideally suited to the BEM. There are three planes of symmetry in the problem. Two of them are genuine planes of symmetry. The mesh shown in Fig. 6 represents only one quarter of the total problem. The third plane of symmetry is put on the sea surface to enforce the boundary condition $\partial u/\partial n = 0$ on the sea surface, which saves having to put elements there. This is the reverse of the normal procedure used in the FEM where a plane of symmetry is represented by the boundary condition $\partial u/\partial n = 0$. There are also some elements not shown in Fig. 6 a large distance from the platform. These were put there to enforce the boundary condition required by the designer of $\partial u/\partial n = 0$ at infinity. Without these elements the BEM would automatically enforce a boundary condition of $u = 0$ at infinity which is not what the designer wanted. It should be noted that this mesh at "infinity" is not connected to the mesh on the structure itself. The smallest elements used in the problem had dimensions of 460 × 45 mm (on the anodes) and the largest had dimensions of 30 × 30 m (on the "infinite" boundary). The problem was run on an earlier version of BEASY which only had constant elements, and was analysed using 653 elements. If it were to be re-analysed today the number of elements could be considerably reduced by using quadratic elements. Several analyses were performed as the designer not only wanted results for the complete system but also wanted to know what would happen if various combinations of anodes were switched off. The ability to restart the analysis at various intermediate points was very useful here.

Rotating Shaft

Fig. 7 shows the mesh required for the stress analysis of an axisymmetric shaft rotating at 3000 r.p.m. Points to note are

i) No elements are required on the axis of symmetry
ii) The mesh refinement around the area of expected stress concentration is easy to achieve.
iii) Body forces such as centrifugal forces can be handled using BEASY.

All the commands needed to create the data are given below.

```
RZ 50
BP 1  120   0
BP 2  120 200
BP 3  120 280
BP 4   60 280
```

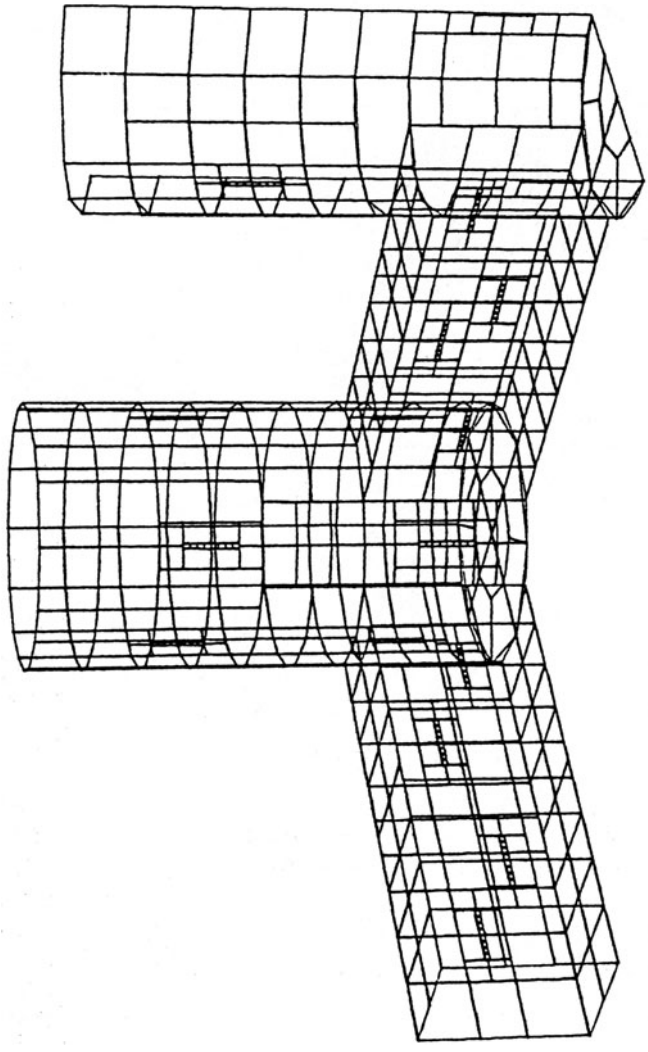


Figure 6 BE Mesh for an Offshore Platform

```
BP 5    60 220
BP 6    40 220
BP 7    40 200
BP 8     0 200
BP 9     0   0
BP10    60   0
BL 1 1 2 5 3
BL 2 2 3 3 3
BL 3 3 4 3 3
BL 4 4 5 3 3
BC 5 5 7 6 4 3
BL 6 7 8 3 3
BL 7 9 10 3 3
BL 8 10 1 4 3
ZE 211000
ZP 0.3
ZD 7.7916E-9
```

Figure 7 BE Mesh for a Shaft

The stresses resulting from this analysis are plotted in Fig.8. The same problem was run using the ASKA Finite Element System and very similar results were obtained.

Turbine Disc

Fig. 9 shows the analysis of transient temperature distributions in a turbine disc. The initial temperature of the turbine disc is 295.1°K and the values of the thermal conductivity, density and specific heat are $5 W/m^{-1}\ °K^{-1}$, 8221 kgm^{-3} and 550 $JKg^{-1}\ °K^{-1}$ respectively. There are 18 different zones along the boundary each with a different set of prescribed values for the heat transfer coefficient and the temperature of the surrounding gas and their time variation at one of such boundary zones is shown in Fig. 9. Note that the mathematical representation of the heat transfer coefficient implies the use of mixed boundary conditions of the type $\alpha u + \beta q = \gamma$. No special difficulty is

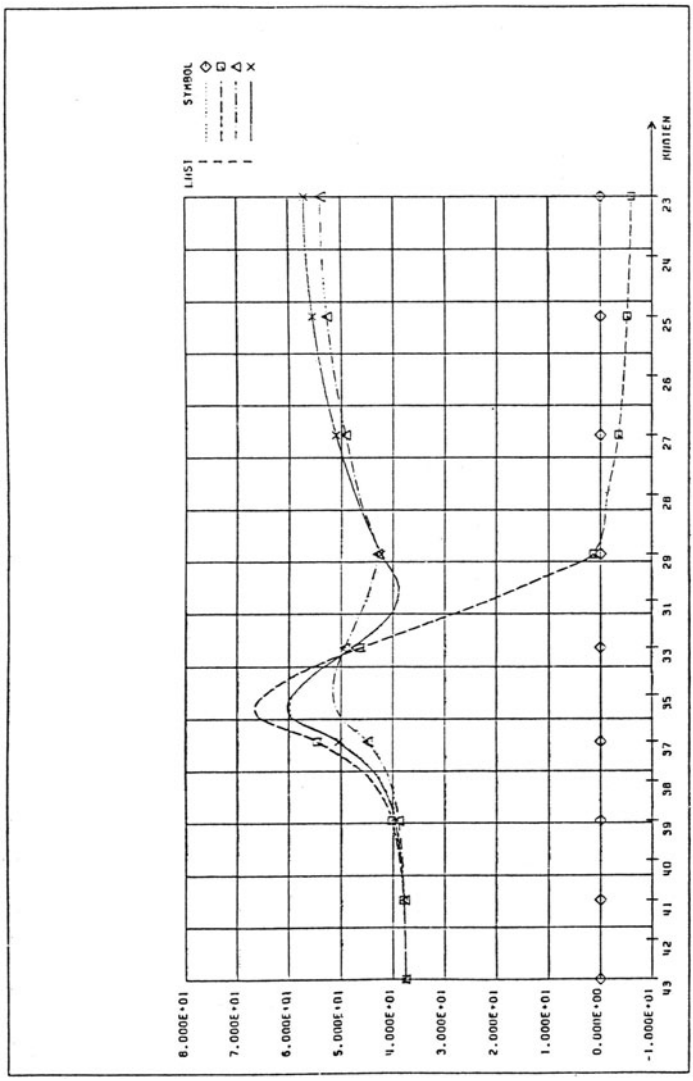


Figure 8

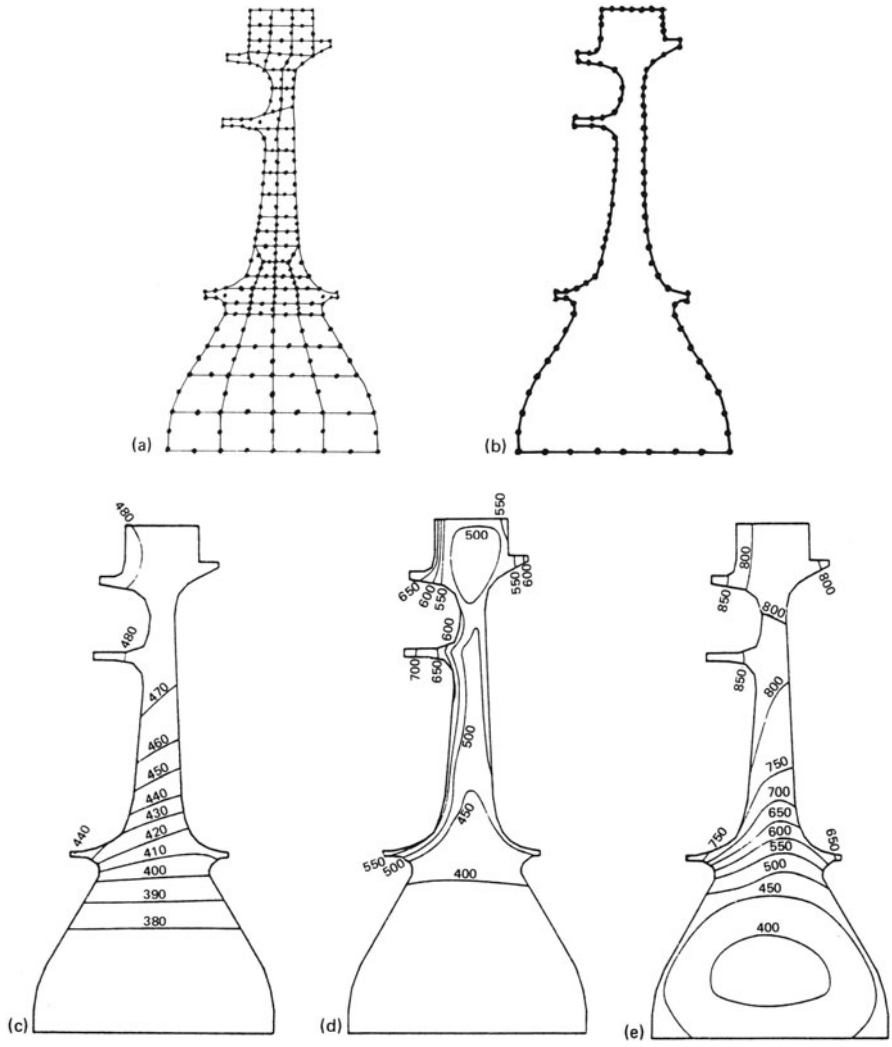


Figure 9 Turbine disc: (a) FEM mesh (b) BEM discretisation (c-e) boundary element results

associated with this implementation.

For comparison purposes a FEM analysis employing 71 quadratic isoparametric elements and 278 nodes was also carried out. The BEM discretization employed 90 linear elements and 106 nodes (there are 16 double nodes to allow for the discontinuities on the boundary data at the intersection of boundary zones). A stepwise linear variation was prescribed for the boundary temperature. For the boundary flux it was assumed to be linear or quasi-quadratic according to the variation of the heat transfer coefficient and external temperature within each step.

Results (isothermals) at a typical time are plotted in Fig. 9 for both numerical methods showing excellent agreement.

It is important to point out that the BEM results were obtained using a novel approach of referring the integral equation always to the initial conditions. As the initial conditions are usually everywhere zero this means that one always is solving a boundary only problem, i.e., only boundary integrals need to be computed. This technique presents important advantages for unbounded domains. The time dependent option of the BEASY program can use internal cells within the domain in the above technique of referring all variables to the boundary. The time dependent codes can be used for the solution of two-dimensional and axisymmetric potential problems.

5. CONCLUSIONS

BEASY is a comprehensive commercial application package of the BEM. It is being used by several large industrial companies who are finding it a useful tool in their design offices.

Its chief advantages over FEM packages are

i) Only having to discretize the boundary greatly eases data preparation. This is in the authors' and users' opinion the greatest advantage of the method. With computing costs still declining and engineers' time becoming more expensive the saving in engineers' time is far more significant than savings in machine time. Also, engineers welcome anything which relieves them of the dreary task of data preparation and leaves them free to concentrate on more important tasks. Even more fundamental is the fact that analysis invariably lies on the "critical path" in the design and production process and any tool which can shorten the "turn-around" time through the analysis office can bring forward the date of completion of the project. This in turn has very significant economic effects, particularly in a competitive world. It is often said that modern mesh generators can make FEM data as easy to prepare as BEM data. However, it is the authors'

opinion, based on our constant visits to industrial companies many of which have invested heavily in FEM packages, that mesh generation is still a major problem.

ii) The ability to handle infinite domains. A surprising number of problems fall into this category and the difficulty of using FEM for this type of problem is obvious.

iii) The reduced number of degrees of freedom needed to analyse a typical problem means that large problems can be handled on small minicomputers.

iv) Results at internal points are obtained only at points requested by the user. As the user is rarely interested in the full field solution (and is in fact often interested only in the boundary solution) this is a distinct advantage.

All these advantages point to the importance of boundary element methods and the need of providing industry with adequate software. Although a few finite element packages have some boundary element facilities BEASY is the first comprehensive boundary element package available to the practising engineer.

REFERENCES

1. BREBBIA, C.A. "The Boundary Element Method for Engineers", Pentech Press, London, Halstead Press, New York, 1978: second printing 1980.

2. BREBBIA, C.A. and WALKER, S. "Boundary Element Technique in Engineering", Butterworths, London, 1979.

3. DANSON, D.J. A Boundary Element Formulation of Problems in Linear Isotropic Elasticity with Body Forces. Boundary Element Methods. Proceedings of the Third International Seminar, Irvine, California, July 1981. C.A. Brebbia (Editor)

4. DAS, P.C. A Disc Based Block Elimination Technique used for the Solution of Non-Symmetrical Fully Populated Matrix Systems Encountered in the Boundary Element Method. Recent Advances in Boundary Element Methods. C.A. Brebbia (Editor) Pentech Press 1978.

5. GROENENBOOM, P.H.L. The Application of Boundary Elements to Steady and Unsteady Potential Fluid Flow Problems in Two and Three Dimensions. Boundary Element Methods. Proceedings of the Third International Seminar, Irvine, California, July 1981. C.A. Brebbia (Editor).

6. TELLES, J.C.F. and BREBBIA, C.A. "Elasto-Plastic Boundary Element Analysis", in "Non Linear Finite Element Analysis in Structural Mechanics", Wunderlich, W., Stein, E. and Bathe, K.J. (Editors), Springer Verlag, 1981.

7. WROBEL, L. and BREBBIA, C.A. "Time Dependent Potential Problems", Chapter in "Progress in Boundary Elements, Volume I" Pentech Press, London, Halstead Press, New York, 1981.

C.A.E. INFORMATION, EDUCATION AND EXPERIENCE

MICROCOMPUTER BASED COMPUTER AIDED DESIGN SYSTEM APPLICATIONS IN ENGINEERING EDUCATION

Bernard S. Charles, Lumbermate Company, St. Louis, Missouri U.S.A. and Phillip L. Gould, Harold D. Jolley Professor and Chairman, Department of Civil Engineering, Washington University, St. Louis, Missouri U.S.A.

INTRODUCTION

Engineering design may be a repetitive, tedious and time-consuming process, and, for these reasons, is an ideal application for computerized automation. Computer Aided Design (CAD) is viewed in industry as a cost effective means of improving engineering productivity, accuracy, and project control. Unfortunately, the use of CAD Systems in Engineering Education has not kept pace with current industrial applications. This is mainly due to the high cost of both hardware and software required for the development of even small CAD systems.

The information presented herein is based upon ongoing research in the Department of Civil Engineering at Washington University in St. Louis, Missouri, U.S.A. There it was demonstrated that CAD software is a valuable teaching aid and that appropriate educationally oriented CAD software can be implemented on low cost microcomputer systems. Such systems may enable engineering schools to acquire representative CAD software within the limitations of academic level budgets.

The software developed at Washington University is used in an introductory structural design course, and addresses such topics as the analysis and design of trusses, continuous beams, and plate girders. Special purpose programs were produced for the inexperienced user and served to provide an accelerated design experience. In this paper the educational benefits of CAD are cited, the programs are described, and the physical environment and configuration of the system are discussed.

EDUCATIONAL BENEFITS OF CAD

The CAD system developed in the Department of Civil Engineering at Washington University in St. Louis provides accelerated

hands-on experience by allowing students to rapidly solve realistic design problems. Relieved somewhat from tedious structural analysis and design calculations, students have the time and freedom to explore many alternatives. Each iteration gives the students practice at making judgmental engineering decisions and helps to impart a feeling for the design process, which they would not otherwise have acquired in such a short period of time. Further, by adjusting the input variables so as to ultimately satisfy a specified criterion, the students acquire some insight into the sensitivity of the controlling equations to these parameters.

It is the opinion of the authors that engineering software should not be used by students, or even practicing engineers, simply as a "black box"; rather, the underlying principles must be understood. Therefore this educational CAD software was designed to enhance, but not to replace classical teaching methods. Each student had at least some familiarity with each design problem using a traditional approach before the computer experience. From this basis, the process of investigating a multiplicity of solutions is thought to reinforce the theoretical concepts.

Educational CAD software is a valuable teaching aid because it provides immediate feedback, and leaves all engineering decisions to the user. A further benefit is that students are exposed to CAD, computer graphics and microcomputers, which are tools which many will undoubtedly encounter and use in their future careers.

DESCRIPTION OF EDUCATIONAL CAD SOFTWARE

As is the case with the development of any interactive software system, it was necessary to evaluate the needs and level of the users. The students are proficient in structural analysis and familiar with basic design concepts, but are in general inexperienced in computer operations. Therefore, the CAD software had to be written with this inexperienced user in mind. This was achieved by providing a separate program for each design topic, and by constructing a "user-friendly" interface for each program. The specialized nature of the programs enabled students to concentrate on specific topics. User-computer interfacing, consisting of menus (the menu for the truss program is shown in figure 1); unambiguous and informative input prompts; easy to understand textual reports; comprehensive input data checking; and color graphics feedback (figures 2 and 3 are graphic displays of cross sectional geometry), comprised eighty percent of the programming effort (both time and lines of code).

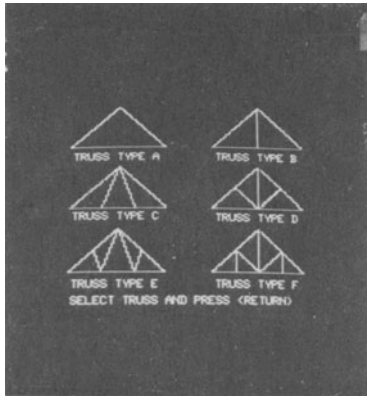


Figure 1.
Display of Truss Types

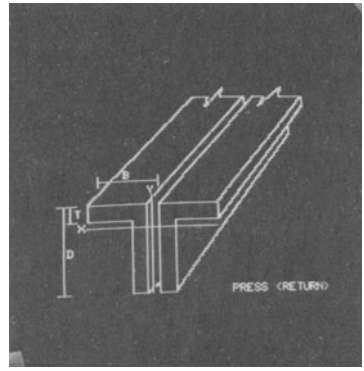


Figure 2.
Cross-Section of Truss Member

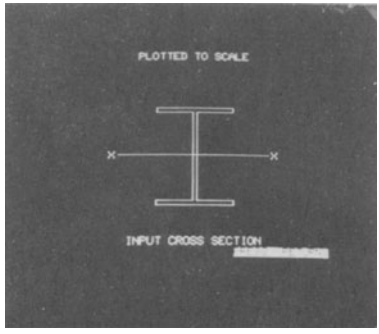


Figure 3.
Cross-Section of Beam

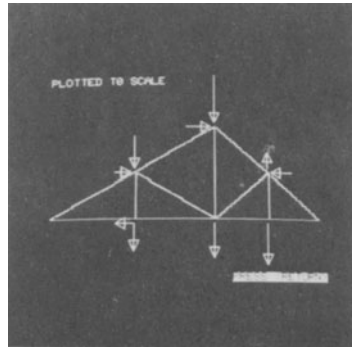


Figure 4.
Truss Configurations

High resolution graphics permitted both easy verification of input data and the graphic display of structural analysis results. Representative displays of structure configurations and loadings generated by the truss, continuous beam, nonprismatic beam, and composite beam programs are shown in figures 4 through 7, respectively. The analysis results for the truss and continuous beam programs are shown in figures 8 and 9 respectively. Analysis results for the nonprismatic beam program are shown in figures 10 through 12.

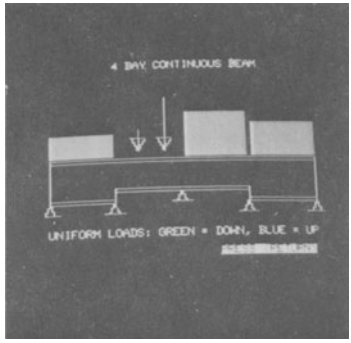


Figure 5. Continuous Beam Configuration

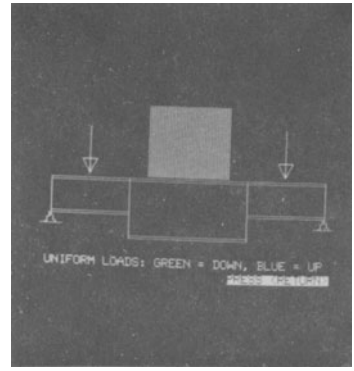


Figure 6. Nonprismatic Beam Configuration

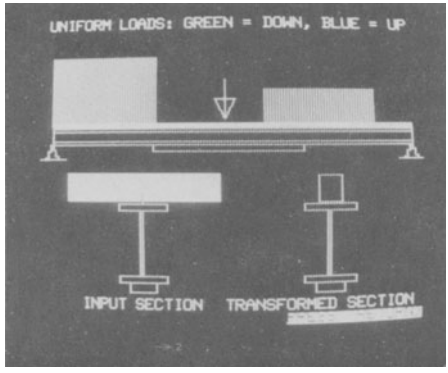


Figure 7. Composite Beam Configuration and Cross-Section

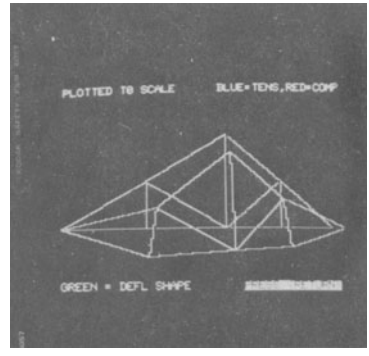


Figure 8. Results of Truss Analysis

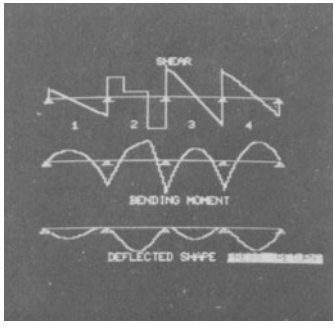


Figure 9. Results of Continuous Beam Analysis

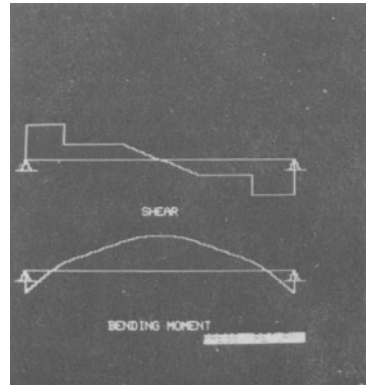


Figure 10. Results of Nonprismatic Beam Analysis

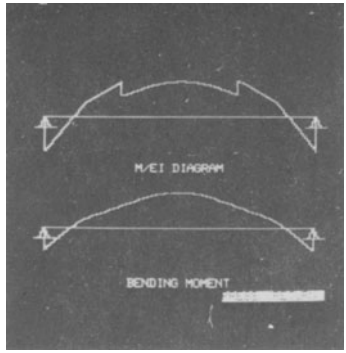


Figure 11.
Results of Nonprismatic
Beam Analysis

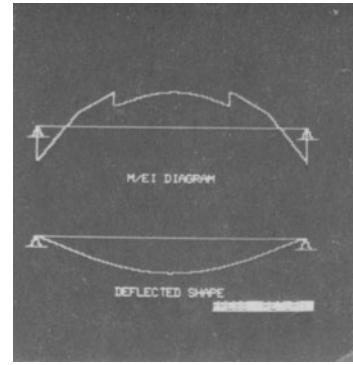


Figure 12.
Results of Nonprismatic
Beam Analysis

Color was used to amplify and clarify analysis results. For example, in one display, tension members were shown in blue, compression members in red and the deflected shape was superimposed in green. There, the use of color enabled the programmer to combine all this in one display, which aids the user in interpreting the results and in gaining some insight into the load resistance mechanism of the truss.

THE MICROCOMPUTER ENVIRONMENT

There are many benefits to working in a single-user microcomputer environment as opposed to a multipurpose, multiuser-situation. The microcomputer user is not troubled with poor response time or long queues to gain access to peripheral devices (printers, plotters, tape drives, etc.). Operation and maintenance costs are low due to the fact that special physical facilities are not required and there are no time-sharing charges. Both program developers and users enjoy the personal, user-in-control environment accompanying the microcomputer. Memory limitation can be a drawback, but the strict memory limitations imposed by many microcomputers can be overcome by selecting an appropriate operating system as discussed in the next section.

MICROCOMPUTER CONFIGURATION - HARDWARE AND SOFTWARE

The software described herein was developed on an Apple II Plus microcomputer. The hardware configuration was 64 kilobytes (K) of random access memory (RAM), two 5-1/4 inch flexible disk drives, which provided a total of 287K of on-line storage, a color video monitor, and an 80 character per second (CPS) dot matrix printer with graphics capabilities. This is the minimum hardware configuration which was deemed satisfactory for our applications and operating system requirements.

The Apple II was chosen because at the time (summer 1980) it was the only low-cost microcomputer which was supported by high level structured programming languages and which had built-in high resolution (280h x 192v) color graphics. Since then, many suitable microcomputers have been marketed. The UCSD p-System was the operating system selected. This is a powerful and easy-to-use menu-driven system designed specifically for the development of software in a microcomputer environment. It is fully integrated, consisting of FORTRAN/77, Pascal, and BASIC language compilers, text editor, file management utility, assembler, link editor, graphics library, and memory management capabilities (chaining, overlaying, separate compilation and dynamic memory allocation).

CONCLUSION

The results over a two-year period have been very encouraging. Students find the environment to be friendly and responsive and the faculty has noticed that many students have come to understand the design process more rapidly than by working in a conventional academic situation.

COMPUTER AIDED ENGINEERING - THE EDUCATIONAL ENVIRONMENT

Wayne E. Carroll

University of Central Florida, CEES Department
Orlando, Florida, USA

ABSTRACT

The advent of inexpensive microprocessor-based computer systems signals the end of an engineering era. No longer can the practicing professional remain competitive using "hand" calculators or "slide rule" technology.

Neither can educational institutes enjoy the luxury of producing a product that operates in this antiquated environment.

This paper discusses the University of Central Florida's response to the impact that this technology is having in the private sector. In particular, the integration of an interaction computer graphic-CAD approach to solving structural engineering analysis and design problems. Examples presented serve to document not only the scope of this effort, but its effectiveness in the context of the educational environment.

INTRODUCTION

The University of Central Florida's Civil Engineering curriculum is fairly typical of what exists at many American universities. Essentially, it contains three years of fairly broad studies in the Engineering Sciences coupled with one year of specialization in the Civil Engineering option. Primary emphasis in the last year of study includes coursework in the areas of hydraulics/hydrology and soil mechanics, as well as the structures discipline. Coursework taken by seniors in this latter discipline include:

1. Structural Analysis
2. Matrix Structural Analysis
3. Concrete Design, and
4. Steel Design

The intent of this paper is to document efforts to integrate an interactive computer graphic approach methodology for solving problems in this area of concentration.

To a large extent, motivation for this approach to education can be directly traced to the impact microprocessor-based computers are having in the Civil Engineering work place. The acceptance of microprocessor-based mini-computers into this environment has produced a need for graduates with not only traditional structural analysis skills, but with the skills necessary to program and provide software maintenance support, as well as having a greater understanding of computer practices as they apply to the profession.

According to the American Society of Civil Engineers, 85% of all U.S. Civil Engineering offices have less than 15 employees. This limited resource base does not lend itself to the approach to system software maintenance available in other areas of the engineering discipline (i.e., automotive and aerospace). Yet, much of the technology currently employed in these latter engineering disciplines apply equally to the Civil Engineer. In particular, the CAD approach to system/component design.

COMPUTER AIDED ENGINEERING - A DEFINITION

Computer Aided Engineering, sometimes referred to as CAD, really has a multilevel definition. In its most sophisticated state, it represents the ability to interact with computers in a fashion that allows a user to take a project from conceptual design to the final working drawing stage. It can include design optimization algorithms and, by necessity, involves large data base management. For manufactured products, a direct interface with CAM can also exist.

This technology just described exists, but still has not matured to the point that it actually does work all the time. At a somewhat lower level, system component analysis and design are performed interactively using computers, but the data base management structure necessary to integrate it with other functions in the design process such as the production of working drawings does not exist.

This latter state probably represents the practical level of computer aided engineering that can be currently expected in the small Civil Engineering office environment, given the cost of today's equipment.

In this paper, discussion will, therefore, center on utilizing this lower state approach to analysis and design in the context of the University of Central Florida's Structural Engineering course offerings. Current plans call for expanding the project to include elements associated with the more comprehensive system approach initially discussed.

PROCESSING ENVIRONMENT

At present, all computer programs developed to support this effort reside on commercially available 4050 series Tektronix desktop computers which contain 32K of working memory. In selecting an appropriate microprocessor, the importance of having an extensive graphics capability was a paramount consideration and, in turn, dictated equipment selection. Although this microprocessor does not feature the versatility available on many microprocessors relative to language options, it offers excellent line editing and extensive syntex checking features. These latter features have been found to be extremely helpful for first-time users. The programming language featured on this system is a very simple version of BASIC which appears to be the most prevalent language currently available on microprocessors. The current laboratory configuration includes a Tektronix 4909 mass storage device that is available to each user.

STRUCTURAL ANALYSIS SOFTWARE

Interactive software has been developed to support the Structural Analysis and Matrix Structural Analysis courses currently taught in the structures discipline. Initially, assignments in the Structural Analysis course focuses on having students develop programming proficiency. Typical assignments usually given during the course duration include:

1. A program to develop shear and moment values for a simply supported beam with uniform and concentrated loads.
2. A program to plot output from assignment #1.
3. An introduction to finite difference techniques with an assignment to develop a program to determine displacements for assignment #1.
4. A program to analyze a statically determinant truss.
5. A program to develop pre-processing graph for assignment #4.

In the Matrix Analysis course, assignments are slightly more extensive in nature and are used to supplement this course content:

1. Convert from FORTRAN a Gauss Elimination Equation Solver.
2. Modify assignment #1 to solve a symmetric banded equation.
3. Solve a series of equations and draw conclusion on memory and time requirements using the program developed in assignments #1 and 2.
4. Write a program to solve a structurally indeterminant truss including pre-processing graphics.

5. Write a program that determines the shear and moment in a beam for uniform and concentrated loads with various boundary conditions.
6. Write a simple program to analyze a frame structure.
7. Using an existing program, analyze a complex frame.
8. Using an existing program, develop a two-dimensional FEM model involving a stress concentration.
9. Using an existing FEM, code, analyze and interpret results.

Figures 1-6 indicate typical forms the assignments take in their completed stage.

DESIGN COURSEWORK

The two design courses taken by students in the structures discipline are Steel and Concrete Design. These courses traditionally examine what will be terms "component" design (i.e., design a beam or column given the loads, etc.). It has been the feeling of the faculty that this approach does not provide the student with an opportunity to fully appreciate the relationship between the analysis and design process. One change resulting from this recognition was the creation of an integration course in both the Steel and Concrete Design areas. In essence, what has been created is a two-course sequence in each area.

The first 4-semester hour course in, say, Concrete Design exclusively focuses on "component" design. The following semester a one-semester hour course is taught in which an actual project is designed and detailed.

Computer software that has been implemented in these courses has been developed along two separate lines of thought. In the first course, which has always emphasized the use of handbook design aids, software has been developed to closely parallel these existing aids. This serves to reinforce the traditional course content, but also demonstrates the application of computers and computer graphics to component design. Figure 7 probably best typifies the implementation of this philosophy.

The use of interactive diagrams similar to the computer generated one shown in Figure 7 is a traditional approach to concrete column design. The use of the computer, however, clearly shortens the design process. Other component software products that have been developed in the concrete courses include:

1. Beam design
2. Footing design
3. Retaining wall design
4. Equivalent frame program

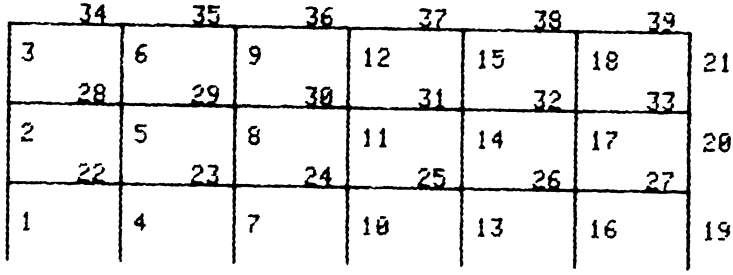


Figure 1. Frame Pre-processing.

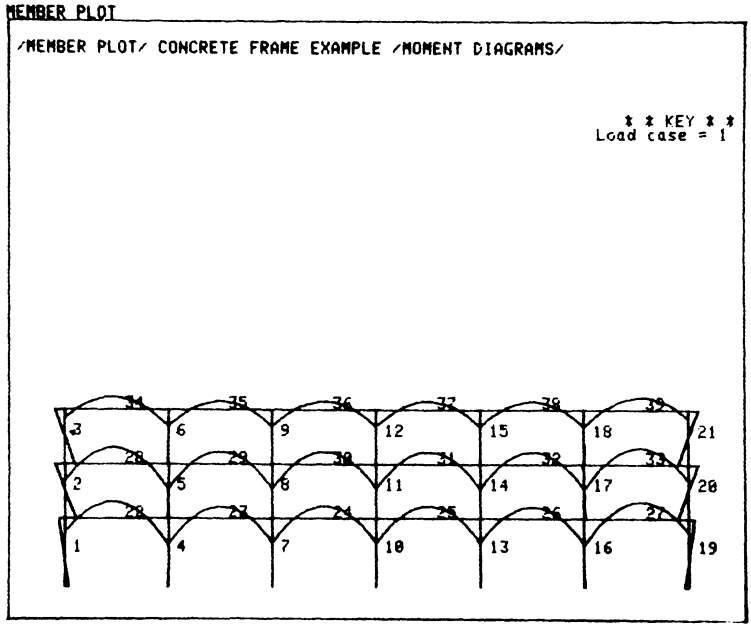


Figure 2. Frame Post-processing.

FACTOR OF SAFETY= 2.47 AT X= 400 Y= 500 R= 475
EARTHQUAKE= 0.01

CIRCLE DEFINITION
X-ORD OF CENTER 400
Y-ORD OF CENTER 500
CIRCLE RADIUS 450

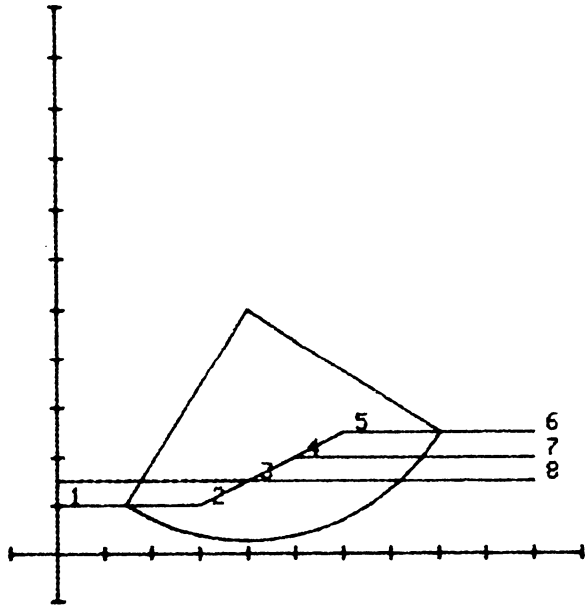


Figure 3. Slope Stability Analysis.

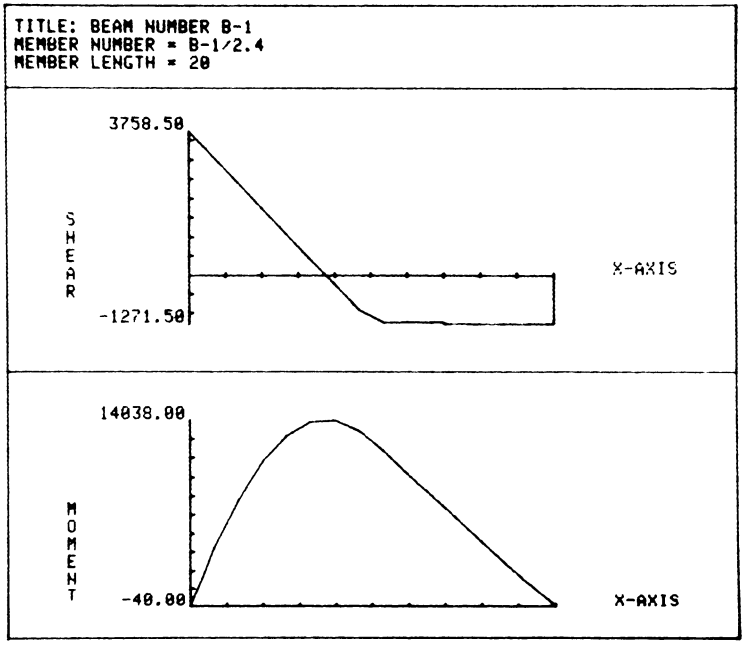


Figure 4. Shear/Moment Analysis.

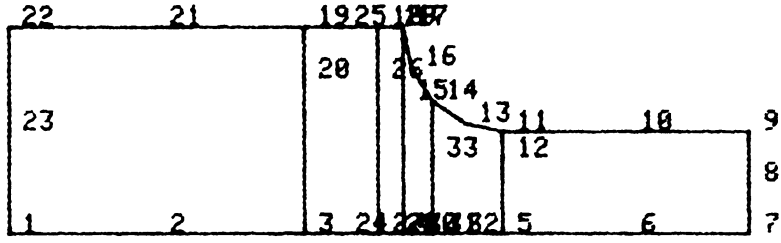


Figure 5. FEM Input Mesh.

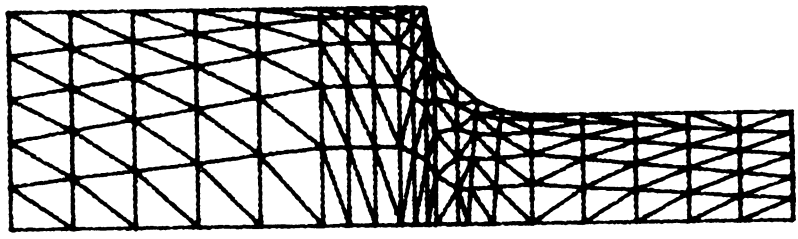


Figure 6. Final FEM Mesh.

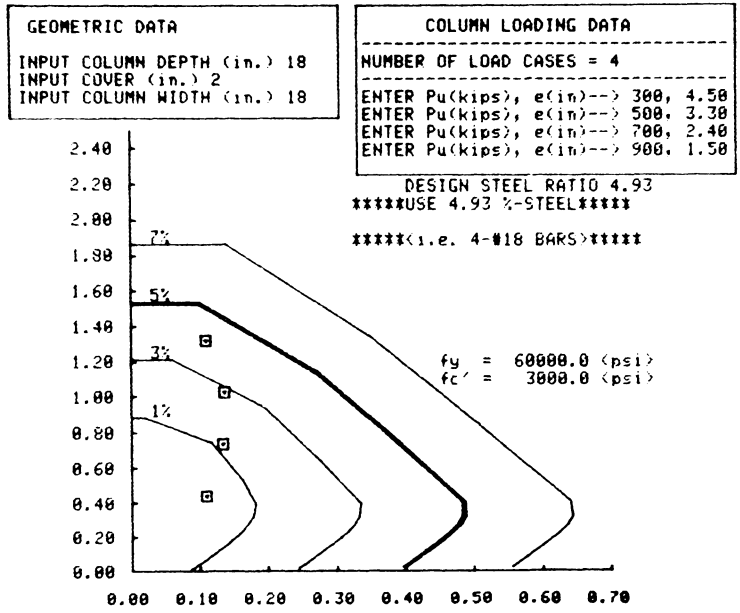


Figure 7. Concrete Column Design.

In the second course, emphasis focuses on integrating the analysis and design courses. Computer programs with a somewhat integrated data base are used to first perform analysis of a project and then perform design. Total integration of the data base is not complete and redundant transfer of data between computer codes is unfortunately still required.

FUTURE DEVELOPMENTS

At present, output from all computer codes must eventually be translated into final working drawings. Software is currently being written to address this stage of design. Figure 8 shows a typical example. Ultimately, all component design software will feature working drawing output, the eventual goal being the integration of the analysis and design process.

CONCLUSIONS

The present paper can only be viewed as a progress report. Response from graduates would indicate the curriculum to be a success. Students seem much more aware of potential computer applications and when faced with employer resistance, have actually purchased their own computers in retaliation. Several former students have also developed significant computer codes for analysis and design in the hydraulics/hydrology discipline, based on their exposure in the structure discipline.

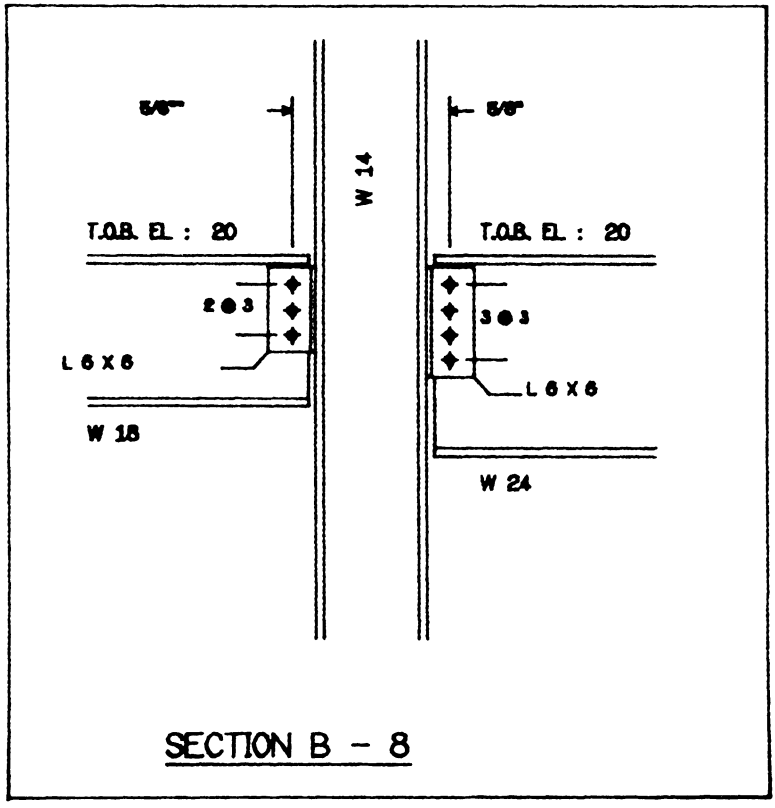


Figure 8. Typical Detail.

THE CLEARING HOUSE FOR ENGINEERING SOFTWARE

B Stanford-Smith

Group Manager - Information & Publications
The National Computing Centre

Before considering computers and engineering, let us look at economics. Computer aided engineering requires investment in new technology, which has to be justified so it is appropriate to consider the business background.

Engineering employs one out of every seven UK employees. According to NCC's National Computer Index it has nearly 20% of the installed computers. Engineering was at the foundation of the Industrial Revolution. Our engineering skill enable us to improve our standard of living, exporting our goods and our expertise. Recently engineering has declined and it is usual to blame the general recession on the assumption that engineering will recover as soon as the recession ends.

However, over the post war years, in British Industry in general the trend has been for unemployment to grow and imports to rise at a greater rate than exports. We are therefore changing from a surplus of income over expenditure to a deficit. Of course inflation tends to make financial figures more alarming but our performance relative to other industrial countries shows that we have lost our position in the league and are still falling.

Of course, various reasons can be put forward for this decline. One is politics, but governments of the left and right have been unable to stop the general trend. Another is automation, although the country with the highest level of mechanisation - Japan - is suffering least in the present recession.

Indeed it is acknowledged that if we are to remain competitive and safeguard jobs that remain, we must both use and supply information technology products. I suggest that one reason why we do not buy British goods is because the design is not as good as its foreign counterpart. Poor design has been one of the contributory factors to our inability to hold our own in world markets. It is reasonable therefore to look to information technology to help us improve design and reduce prices.

INFORMATION TECHNOLOGY

Information Technology is the name given to the congruence of computers, communications and microelectronics. It brings the second industrial revolution whose effects will be at lease as significant as the first. A simple definition of Information Technology is the capture, movement, processing and presentation of information of all kinds. We can now put 10,000 components on a single chip and by 1990 we shall be able to put 10 million. We can store 10 billion characters - enough for a library in a space the size of a desk drawer. Our normal telephone lines will transmit 10 million characters every second and fibre optics, satellite broadcasting and cable TV offer much greater capacities. A British machine has a 2,000 word vocabulary of commands. Plasma and liquid crystal technology will replace all bulky cathode ray tubes with flat screens by 1990 so that small hand-hold receivers will be readily available. If we are to remain an advanced nation then we must be part of Information Technology: its development, its supply and its use.

THE FEASIBILITY STUDY

Three years ago the Mechanical Engineers and Machine Tools Requirements Board became aware that engineers were not taking readily to CAD/CAM. This could have been put down to normal British reticence to accept new ideas but the Board were more charitable both in thought and in deed in that they believed that a major handicap to the widespread use of engineering software was because engineers were unaware that programs existed to help them in their work. There was no single source which knows what systems are available and who supplies them. Accordingly, the Board commissioned a working group which recommended that a clearing house should be set up where suppliers could place information on their software and prospective users could make enquiries.

The main emphasis of dissemination was via a manned enquiry unit. Other methods discussed were publishing, on-line, viewdata but the main justification of the service was for enquirers paying £30 a time to learn if software was available and suitable for the task they were tackling.

The report was kind enough to recognise the achievements of The National Computing Centre in the area of Information on computing. NCC ran an enquiry service and naturally enough one of its databases and areas in which it accepted questions was software. The working group 's report went beyond commending NCC's experience in computing and information handling and praised their achievements with marketing.

NCC COMPUTING INFORMATION

NCC have had an Information Service since 1970 and have developed and maintained six databases on computing topics: hardware, installations, literature, services, software and suppliers. An on-line system was developed in 1972 utilising a thesaurus of computing terms that grew to 8,500 linked entries. The main outlet was the Answering Service that deals with over 4,000 questions a year.

At the time of the Board's investigation, the Information Service was undergoing a drastic reappraisal as to its method of operating. Not so much because of the usefulness of the service but because of its cost. NCC was finding that the wish to answer any question on computing is an expensive exercise. Similarly, was realising the truth in the saying that nobody pays for information especially when they become used to receiving it for nothing. NCC's Information had a history of being available for nothing because it enjoyed government funding. However a change in government policy brought in Boards who provided support for specific desirable tasks and not on-going activities.

With this change in its financial position, NCC turned its attention more closely to those commercial areas that were not self-supporting such as Information. Accordingly there were plans to change the situation by eliminating expensive tasks, producing a range of products and introducing a new enquiry handling system. The working group were aware of these moves, felt that the suggested method of working was sound and recommended that NCC should be invited to tender for the Clearing House for Engineering Software.

THE SUBMISSION

NCC's proposal in turn accepted the need for a clearing house but it tended to play down the importance of the enquiry side and place greater emphasis on publishing.

NCC's own experience was that enquirers were reluctant to appreciate the value of one answer but were more willing to pay for a reference work in which they could do their own searching.

The proposal tended also to reject dissemination by electronic means which the report had raised as a possibility. Again it was felt at that time that the costs of electronic publishing might not be covered by the income. The Board were looking for viable projects. They were restricting themselves to pump-priming exercises and similarly, NCC Information was only interested in work that led to commercial products. The view on electronic dissemination may be open for reconsideration today because technology has moved faster than expected and because there was an unforeseen delay in starting the project. It was presented in October 1980 but a contract was not issued until November 1981 - over a year later.

THE PROJECT

Work started therefore in January 1982 on the basis that a database would be created on engineering software and that this would be made available in one of three ways:

A quarterly newsletter
An annual directory
An enquiry service

It was decided to use a dedicated microcomputer and the choice fell on a British machine: the Trivector Four marketed by Alpha Business Computers. This has the BOS operating system supporting Autoclerk and Autowriter software that gave the machine the capability of database handling and word processing; the latter being required for text handling. NCC's own software database had been virtually all numeric or parameterised. Although sensible theoretically for software descriptions and comparisons, it had been found in practice that parameters cannot describe everything and that tables of facts are uninviting to the user. Appearance and user attitude are factors that have to be part of design.

The Trivector configuration included a matrix printer, the usual screen and a 22 M Byte disc. The working group felt that there were at least 1,000 CAD/CAM packages and that this was a growth area.

COLLECTION

Having selected the vehicle, it was necessary to design the passenger and the record contents were specified and translated into an input form. The requested items included acroynm, equipment, operating system, input, output, availability, price, a 100 word description and of course suppliers name, address and telephone.

One warning! It is easy at this stage to try to put too much into a record and it is easy to think up other descriptions which could be imagined to be of prime importance to a prospective user. However a long record

1 Diminishes the chances of getting your input form completed and returned.

2 Increases the chance of an error occurring or the record becoming out of date.

3 Increases the cost of input handling and printing.

In most cases a buyer is satisfied with just a minimum of information on which to make an initial quick assessment and cut down an array of products into a manageable shortlist of possibilities. He usually wants to be pointed in the right direction in order to make his own detailed appraisal. There is therefore the danger that a lot of the cost and effort will be spent maintaining information that is little used and whose cost is rarely appreciated - but whose chance of error is greater and likely to cast aspersions on the whole database.

Once the input form contents were finalised, it was first sent as a trial to ten suppliers whose views contributed to the final version. The next step was to find an initial mailing list of suppliers who were sent an explanatory and imploring letter together with the form. Progress as regards returns and thereby entries to the database has, it must be admitted,been slow. A considerable amount of follow-up has been necessary - far more than was anticipated. It is a common fault of database controllers that they imagine that all those with potential entries are clamouring to get on.

In truth, everyone is busy and form filling is just another extra chore. A further reason to keep it short and keep it simple.

In this case, it was decided that follow-up should be more effective by phone. And good job too. It was found that in many cases, the named contact was no longer there, for whatever reason I leave to imagination, and a further letter would have stood about as much chance of being replied to as its predecessor. Another interesting point is that where the name we had was not available for whatever reason, it took considerable thought and time for our call to be redirected. We had the feeling that all CAD/CAM software was written in large organisations who were predominantly engaged in other areas and where the CAD/CAM expert was hidden, or was hiding, in some forgotten corner. A completely erroneous situation I don't doubt but an impression all the same.

Having located an authority it usually required further forms to be sent and the follow-up procedures to be started all over again. Meanwhile back at the desk, further likely suppliers are found by scanning the now sizeable CAD/CAM press and looking down lists of exhibitors. This effort has increased the suppliers approached to 400 of whom 205 have so far replied.

MAINTENANCE AND ENQUIRY

However the creation of a databank is not likely to be so tricky as its maintenance which requires considerable checking and cross-referencing. Suppliers expect to see what information you already hold on them and although this is more difficult, experience has shown us that people are more willing to correct your 'mistakes' than start all over again.

Despite the problems, satisfactory progress has been made with the result that we opened the Enquiry Desk for business one month ahead of schedule. The Enquiry system closely follows the new system that was used for NCC's Computing Information and had been running successfully for over two years.

The previous on-line system tended to produce too much information and because of a backlog that searching produced, there were delays in replies. Now the replies are, we believe, brief, relevant and by return. We have in fact received one complaint from an information officer who preferred the old

printout because she could get a supply of coffee and spend a morning poring over it.

The system now works on four levels. The lowest is a catalogue of the subjects covered. In the case of CAE, this is the application areas under which the packages are classified on the input form and which is in the form of a matrix. This catalogue also shows the volume or pages of material available at the next highest level where fact sheets are devoted to each application. The reason for this is that there is a unit charge of 50p a sheet which covers compilation, locating, copying and postage. The total charge is therefore 50p times the number of sheets plus, the one exclusion, VAT.

Fact sheets give brief details of software. In this case, the name, brief text, supplier, telephone and name of contact. In this way an enquirer should have a complete list of software suitable for his task and can quickly eliminate unsuitable programs.

Further information on the remaining short list can be obtained directly from the suppliers or the next highest level (3) of the enquiry service - by quoting the NCC reference number which is unique to each entry. This will provide a fuller description of a chosen package on two sheets of printout at twice 50p or £1 a record.

A further refinement is the final fourth level comprising manufacturers literature for which four sheets are presumed to give another charge of £1 (in this case 25p a sheet to cover copying etc but excluding compilation as it is not originated by NCC).

ADMINISTRATION

Payment can be made by cash with request or by a voucher system. £1 vouchers which can be attached to the enquiry form are available in sheets of 20. The low charge cannot tolerate invoicing but those who have a volume of enquiries may purchase a £100 value module, for a discounted price of £80. The enquiry officers will then maintain an account of usage. This, apart from being cheaper enables enquiries to be sent by phone or telex because no payment need accompany the request.

The system is simple in operation, simple in charging and supplies information in easily digested amounts.

The enquirer knows what information is held (the catalogue) and can obtain just the degree of detail he wants at any stage.

From our point of view, the enquirer defines his requirements and we do not require an engineering expert tied up waiting for enquiries to come in.

Of course the enquirer may have other ways in which he wishes to reach software. He may know the software name or may want to know all software that runs on a particular piece of equipment, or that supplied by a certain firm or within a price range. In this case he can define a search which will be undertaken for £30 an hour (which incidentally and coincidentally is the price of a single enquiry postulated in the working party's feasibility study). Trivial enquiries such as addresses of suppliers are not charged provided they can be answered immediately. Other extensions of the system are fact sheets for CAE on literature, consultants, hardware etc. which for the most part are already on our computing databases.

DIRECTORY AND NEWSLETTER

The other outputs mentioned were an annual directory due during 1983 and a newsletter. NCC has published directories of computing hardware, software, services and suppliers for the last two years and experimented both with conventional printing and laser printing. NCC has of course other publishing experience with a booklist of over 100 titles including manuals and reports as well as conventional technical books.

The newsletter whose first edition appeared in December has three parts: abstracts of CAD/CAM articles, news items(summarised from press releases) and the latest additions to the database. In all these three areas NCC has previous experience in that it produces as separate publications journal abstracts, a news roundup and changes to its installation database.

The end objective of the clearing house is to contribute to the improvement of the design and manufacture of British products. It is to be hoped that the design and production of the database itself is of the highest quality so that it shows the way both as an example and as an indicator to where suitable help is available.

CIVIL ENGINEERING

MODELS FOR THE PREDICTION OF FORCES INDUCED IN OFFSHORE PLATFORM CRANES DURING LIFTING

James A. D. Balfour

Heriot-Watt University

ABSTRACT

This paper describes a computer program capable of predicting the forces induced in an offshore platform cranes while off-loading from supply vessels. The program, which is based upon the finite element method, is shown to be flexible, yet easy to use. Various computational techniques that save processor time and storage are described, along with a strategy to deal with system non-linearities. It is shown that by adoption of such techniques any reasonable two dimensional idealisation can be catered for on a mainframe computer of moderate size. The capabilities of the program are demonstrated by using it to investigate the optimum point of lift-off on the wave cycle and to predict the forces in a small scale physical model during a main hoist lift.

INTRODUCTION

All crane operations are dynamic in nature. Even on land the fact that the load must be accelerated from rest to the lifting speed of the crane means that forces greater than the static forces must be induced throughout the crane structure. On land the lifting speeds tend to be low and the consequent dynamic amplification is small. By contrast, cranes on offshore platforms suffer much more severe dynamic loading. From the moment these cranes are installed they are responsible for the transfer of seaborne supplies to the platform. Disruption to offshore schedules is costly, and consequently these cranes are often expected to operate in high seas.

Under these conditions the load to be lifted is not stationary, but rides up and down with the supply vessel that carries it. The task facing the crane operator is to initiate the lift such that the load leaves the supply vessel as it approaches a peak of its motion; hence reducing the shock

loading on the crane and minimising the danger to the supply vessel and its crew. However, gauging the supply vessels motion from a vantage point almost vertically above is difficult and misjudgement will mean that the crane will lift a downward moving load. Consequently the crane will be subject to high shock loading and the load will take longer to clear the vicinity of the supply vessel, thus increasing the danger to it and its crew.

While offshore oil and gas exploration was confined to relatively calm waters typified by the Gulf of Mexico and the Middle East, the dynamics of offshore lifting posed no practical problem. When, later, exploration moved into the rougher waters of the North Sea the frequency of crane accidents reached alarming levels. A large proportion of these accidents were directly attributable to dynamic loading, with damage being caused to booms, cables and pedestals. In one instance a crane was pulled bodily off the platform and many accidents involved injury and loss of life.

In recognition of the dynamic nature of offshore lifting cranes operating in the North Sea are now subject to derating rules which set set safe working load according to the prevailing sea conditions. The objective of any form of derating is to allow the best possible use of craneage while maintaining acceptable safety levels. To establish a realistic derating criterion is a difficult task, requiring account to be taken of both crane dynamics and supply vessel motion. Any rational derating scheme will produce a different derating factor for each combination of crane configuation, supply vessel and sea state. Unfortunately there is little consistency between derating rules currently used in the North Sea, where safe working loads for given conditions vary widely from one sector to another. Inspite of heavy derating, accident statistics show that serious accidents and dangerous occurences continue at unacceptably high levels. The ability to predict the forces induced in the crane structure during lifting is essential for the efficient design and derating of cranes destined for use in exposed waters. A considerable amount of work has already been conducted to investigate a variety of analysis techniques varying from simple mathematical models capable of explicit solution (Charret and Hyden-1976, Johnson-1976, Stenhouse and Hilton-1978) to the use of complex finite element packages and analogue computers (Wyon and MacKinnon-1975, Kenney-1978, Clarkson and Kenney-1980, Strengenhagen and Gran-1980, Watters, Moore and Gill-1980). In a previous paper (Balfour and Owen-1980) the author presented results from mathematical models of varying complexity and argued strongly that purpose written computer programs offer substantial advantages over both simple mathematical models and general purpose finite element packages. A later publication (Balfour and Bowcock-1982) showed these models to be capable of accurately predicting the response of small scale physical models of offshore platform

cranes. That paper also outlined a computational strategy capable of dealing with system non-linearities in an efficient and flexible manner.

The remainder of this paper describes a computer program that is based upon a mathematical model considerably more complex and general than any considered in the previous two publications. The program is capable of considering the complete crane structure in two dimensions and is written, not as a research tool, but in commercial format, thus demonstrating the power and ease of use of a purpose written program. The program`s flexiblity is demonstrated by using it to investigate the importance of initiating the lift near the optimum point on the wave cycle. In addition forces induced during main hoist lifting using a small scale physical model are predicted and compared with the measured response.

CRANE STRUCTURE IDEALISATION

Most cranes used on offshore installations are of the non-counterbalancing, pedestal mounted, high "A" frame type (see Figure 5). Such cranes take up a minimum of valuable deck space. They are fitted with a multi-part main hoist and a single-part whip hoist which is capable of lifting light loads at speeds high enough to allow operations to continue even in moderately severe seas. The boom is normally long to enable the supply vessel to stand well off the platform. Typical capacities are around 70 tonnes for the main hoist and 5 tonnes for the whip hoist. Line speeds tend to be around 1.5 metres per second and 35 metres is a typical boom length. There are a number of ways to idealise these cranes and Figures 1, 2 and 3 show three of them. Mathematical model based upon these relatively simple idealisations are described in earlier publications (Balfour and Owen 1980 and Balfour and Bowcock 1982). The computer program CRANES described in this paper can deal with any idealisation within the range indicated by Figures 4 and 5. In general a more detailed idealisation will better represent th behaviour of the crane and will yield information about motions and forces in crane components neglected by simpler idealisations. Of course more complex models demand more computer storage and processor time.

DESCRIPTION OF THE PROGRAM "CRANES"

General

CRANES is modular in structure consisting of a series of FORTRAN IV routines which form three distinct sections, namely; the front end processor which reads, validates and echoes the input data; the analysis section which solves the problem, and the post processor which selectively outputs results. These three sections can be overlaid, run as separate programs or run as one large program. The optimum arrangement would, of course, depend upon the user and the local computer facilities.

Assumptions

CRANES has been formulated to be as flexible as possible while keeping the number of assumptions employed to a minimum. Currently the assumptions listed below apply.

1. With the exception of the hoist and boom hoist cables the crane structure is represented by constant stiffness beam elements.
2. All mass has been lumped at the node points. (Research has shown that using distributed mass has a computational penalty for no significant improvement in accuracy.)
3. The platform to which the crane is attached is stationary. (Work is in hand, however, to allow base motions, thus catering for cranes on tension leg platforms and semi-submersibles)
4. The load is in the plane of the structure and is moving vertically.
5. The supply vessel rides upon a sinusoidal wave and describes exactly the motion of the water surface. (The modification required to allow any supply vessel motion would be trivial. It is the authors opinion, however, that this would be of no practical benefit)

Capabilities

The principal facilities offered by CRANES are outlined below.

1. By using standard finite element techniques platform cranes of any geometry can be modelled.
2. Element masses are calculated automatically from cross sectional areas.
3. Additional mass can be added at any node.
4. Non-linear cable stiffnessses can be used. (Cables are assumed to have no compression stiffness)
5. Any velocity/time relationship can be used for line retrieval by both the hoist and the boom hoist winches. By allowing the boom hoist winch to operate during the lift, the effects of "booming in" can be investigated.
6. The boom hoist cables run from any node on the cab or "A" frame structure, to any boom node.
7. The hoist cable(s) run from any node on the cab or "A" frame structure, to any boom node, and then to the load.
8. The supply vessel being off-station is catered for by allowing lifting with an off-lead angle.
9. Main hoist lifting is facilitated by allowing independent variation of the number of parts of line between the "A" frame and boom and, boom and load.
10. The lift can start at any point on the wave cycle and the behaviour prior to the load lifting off the deck is accurately modelled.
11. Data preperation is reduced by describing the crane with the boom in the horizontal position. CRANES automatically transforms nodal coordinates for the current boom angle.

Input

Modules The input data is read as a series of modules. Each module starts with a module header card which is recognised by having a * as the first non-blank character. The module is identified by the next four characters, and the module header is followed by one or more lines of data. There are currently nine mandatory modules and five optional modules. A brief description of the modules is given below.

Mandatory Modules

*TITLE	- Job title.
*NODES	- Nodal coordinates.
*ELEMENTS	- Element connectivity.
*PROPERTIES	- Element propeties.
*BOOM HOIST	- Details of boom hoist cables.
*HOIST	- Details of the hoist cable(s).
*INTEGRATION	- Integration and damping parameters.
*CASE	- Details of the initial conditions.
RUN	- Execute, no more data to follow.
*RUN+	- Execute, more data to follow.

Optional Modules

*ADDED MASS	- Magnitude and location of added mass.
*E.BOOM HOIST	- Stress/strain pairs for boom hoist cables.
*W.BOOM HOIST	- Time/line speed pairs for boom hoist winch.
*E.HOIST	- Stress/strain pairs for hoist cable.
*W.HOIST	- Time/line speed pairs for hoist winch.

The modules can be presented in any order except the *RUN* or *RUN+ module which must come last. On encountering a *RUN* module CRANES first checks that all mandatory modules are present and that all modules are error free. It then solves the problem, writing output to disk file. Should *RUN+ be used instead of *RUN* then, after solving the problem described by the current set of modules, CRANES returns to read more modules. These modules could be new (optional) modules or replacement modules. Any module not overwritten by a replacement module is retained, thus enabling a number of problems to be solved as one job without presenting a complete data deck each time.

Comment Cards Comment cards are recognised by having a C as the first non-blank character.

Numerical Data All numerical data is initially read as characters and subsequently post-processed by CRANES. Numbers can take all of the usual forms and are separated by commas and/or blanks.

Errors CRANES has extensive error checking facilities and will output error messages where appropriate.

Data Deck Appendix 1 shows a typical data deck for CRANES.

Solution

Mathematical Background Starting from initial known conditions CRANES uses the popular Newmark Beta Method (Chan, Cox and Benfield-1962) to solve the equations of motion in the time domain. Essentially the known conditions at times "t - Δ t" and "t" are used to predict the unknown displacements at time "t + Δt"; where "t" is the current time and "Δt" is a short interval of time.

By using Newmark`s relationships for future velocity and acceleration the following relationship can be derived.

$$[M + \tfrac{1}{2}\Delta tC + \beta\Delta t^2 K_{s+1}]X_{s+1} = \Delta t^2[\beta P_{s+1} + (1-2\beta)P_s + \beta P_{s-1}] + [2M - \Delta t^2(1-2\beta)K_s]X_s - [M - \tfrac{1}{2}\Delta tC + \beta\Delta t^2 K_{s-1}]X_{s-1} \quad (1)$$

where M is the structural mass matrix.
K is the structural stiffness matrix.
X is a column vector of structural displacements.
P is a column vector of applied forces.
Δt is the time interval.
β is Newmarks Beta. (normally chosen as 0.25)
s-1, s, s+1 subscripts indicating conditions at times t-Δt, t, t+Δt respectively.

Simultaneous Equations At each time step Equation 1 reduces to a classical set of linear simultaneous equations.

$$[LHS]X_{s+1} = [A] + [B]X_s - [C]X_{s-1} \quad (2)$$

$$= [RHS] \quad (3)$$

Where [LHS] is a square coefficient matrix and [RHS] a column vector. For non-linear systems the stiffness matrices in Equation (1) will not be constant. This means that each time step must start with the coefficient matrix [LHS] in unreduced form. Consequently the solution of the simultaneous equations will make heavy demands upon computer time. In a previous publication (Balfour and Bowcock-1982) it was shown that this problem could be avoided by treating the cables as external forces, rather than structural elements. This moves the non-linearities from the [LHS] coefficient matrix to [A] on the right hand side. Hence the coefficient matrix is rendered constant allowing reduction prior to the first time step.Use of the reduced form thereafter shortens the solution time by at least two orders of magnitude for large problems. In addition the treatment of line retrieval by the winches, load pendulation and behaviour prior to lift-off are all simplfied.

Of the reduction methods investigated, Gauss-Jordan inversion without row interchange proved to be the most straightforward and efficient.

Storage Treating cables as forces and inverting the now constant [LHS] matrix means that at each time step the future displacements are obtained by evaluation of the matrix equation below

$$X_{s+1} = [LHS^{-1}] [A + BX_s - CX_{s-1}]$$

where $[LHS^{-1}]$, [B] and [C] are square matrices of dimension equal to the number of degrees of freedom. Taking 70 as the maximum number of nodes likely to be required to describe the crane, results in 209 degrees of freedom (2 for the load, plus 3 for each node, except the pedestal base which has none). Now to store a 209x209 matrix using single precision on a mainframe computer requires approximately 43K words of storage. If all elements of the three square matrices are held the total storage requirement could exceed the storage capacity of some machines. Fortunately matrices [B] and [C] are banded and symmetrical, and by storing only the semi-band of these matrices the storage requirement is reduced by more than 74K (this figure is based upon a maximum band width of 53 which is easily achieved by sensible node numbering). In addition using only the band in matrix multiplications involving [B] and [C] results in significant savings of processor time. Hence by taking advantage of the banded nature of [B] and [C] all reasonable crane idealisations in two dimensions can be handled by a moderately sized main frame computer, without having to reduce the problem to master and slave degrees of freedom.

Processor Time Figure 6 gives an indication of typical processor times for various numbers of time increments and nodes. These times are for the solution phase only on a Honeywell Level 66 machine.

Output
During the solution phase CRANES dumps out the values of the primary unknowns (the displacements) to backing disk store. This allows the results to be selectively post-processed, thus avoiding the problem of having volumes of unwanted output. Post-processing has the additional advantage of enabling the solution phase to run in batch mode and the results to be processed interactively. As CRANES is currently being used in a research environment the post processor is graphics oriented, making use of the GINO graphics library. The program is menu driven and offers the following options.

1. Draw the structure (see Figures 4 and 5).
2. Plot or print the hoist or boom hoist tension (see Figures 8 and 9).
3. Plot or print any displacement.

4. Plot or print any stress resultant (see Figure 11).
5. Plot or print any stress.
6. Plot hoist or boom hoist stress/strain curve.
7. Plot hoist or boom hoist winch velocity profile.
8. Print peak stress resultants for all elements.

As the post-processor is relatively simple it can easily be tailored to meet the needs of the user. For instance to implement a routine that would automatically stress-check every element against some built-in specification would not be difficult.

RESULTS

To demonstrate the flexibility of CRANES, results are presented which show the program being used to tackle two very different problems.

Optimum Point of Lift-Off

Firstly CRANES has been used to investigate how the position of lift-off on the wave cycle affects the dynamic amplification of forces. Figure 7 shows how the maximum tension in the boom hoist cables varies with the lift-off position. It is interesting to note that minimum dynamic amplification occurs at about zero degrees (i.e. when the load has maximum upwards velocity) and not at around 90 degrees, as is commonly thought. Figures 8 and 9 show the time histories of the boom hoist tension when the hoist cable first goes taut at 0 and 90 degrees respectively.

Main Hoist Lifting

For the second problem CRANES has been used to predict the forces in a 1:20 scale model lifting a 1.25 kilogram weight with a six part main hoist. The load is lifted from a horizontal surface which is moving vertically, thus simulating the supply vessel motion. Lack of space limits the presentation of results from this exercise to Figures 10 and 11 which show the characteristic good correlation between the physical and mathematical models. It can be seen that even main hoist lifting, where the hook speed is much lower than for whip hoist lifting, produces significant dynamic effects. This was also seen to be the case when the program was used to investigate main hoist lifting by full sized cranes which showed dynamic effects to be important even in relatively calm sea conditions.

CONCLUSIONS

1. A flexible, yet simple to use computer program, based upon the finite element method, can be developed to predict the forces in offshore platform cranes during lifting operations.

2. For two dimensional idealisations the program can run efficiently on moderately sized mainframe computers provided appropriate computational techniques are used.

3. Such a program should lead to safer crane design and more realistic derating rules.

ACKNOWLEDGEMENTS

This research program was supported by a grant from the Marine Technology Directorate of the Science and Engineering Research Council (UK).

The author also wishes to thank Dr D.G.Owen who initiated this research at Heriot-Watt University and Mr A.O.Bowcock who conducted the experimental work.

REFERENCES

Balfour, J.A.D. and Owen, D.G. (1980) Dynamic Behaviour of Offshore Platform Cranes. Proc., 1980 Offshore Technology Conf., Houston, Texas, Paper OTC3794, pp 509-517

Balfour, J.A.D. and Bowcock, A.O. (1982) Dynamic Response of Offshore Platform Cranes Using Physical and Mathematical Models. Proc. Third Int. Conf. on the Behaviour of Offshore Structures, MIT, Boston, Vol 2, pp 551-560.

Chan, S.P., Cox, H.L. and Benfield, W.A. (1962) Transient Response of Forced Vibrartions of Complex Structural Mechanical Systems, J. Roy. Aero. Soc., 66, pp 457-460.

Charret, D.E. and Hyden, A.M. (1976) Dynamic Factors for Offshore Cranes. Proc. 1976 Offshore Technology Conf., Houston, Texas, Paper OTC2578, pp 565-576.

Clarkson, J.A. and Kenney, F.M. (1980) Offshore Crane Dynamics. Proc. 1980 Offshore Technology Conf., Houston, Texas, Paper OTC3793, pp 497-597.

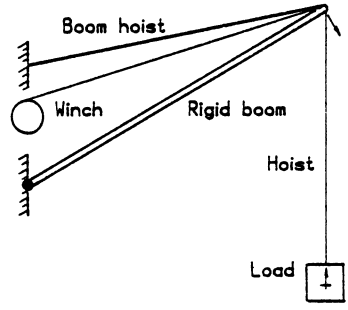
Johnson, K.V. (1976) Theoretical Overload Factor Effect of Sea State on Marine Cranes. Proc. 1976 Offshore Technology Conf. Houston, Texas, Paper OTC2584, pp 601-609.

Kenney, F.M. (1978) Operational Dynamics of an Offshore Crane. Proc. 1978 European Offshore Technology Conf. and Exibition, London, Paper 69, pp 109-116.

Stenhouse, G.A. and Hilton, R.J. (1978) Method of Rating High Pedestal Cranes for Lifting from Supply Boats in Various Sea States. Proc. 1978 European Offshore Technology Conf. and Exibition, London, Paper 71, pp 125-132.

Strengenhagen, J and Gran, S (1980) Supply Boat Motion, Dynamic Response and Fatique of Offshore Cranes. Proc. 1980 Offshore Technology Conf., Houston, Texas, Paper OTC3795, pp 519-534.

Watters, J.A., Moore, D.J. and Gill, I.S. (1980) Dynamics of Offshore Cranes. Proc. 1980 Offshore Technology Conf., Houston, Texas, Paper OTC3792, pp 485-496.



FIGURES 1 AND 2 MODELS WITH ONE AND TWO DEGREES OF FREEDOM

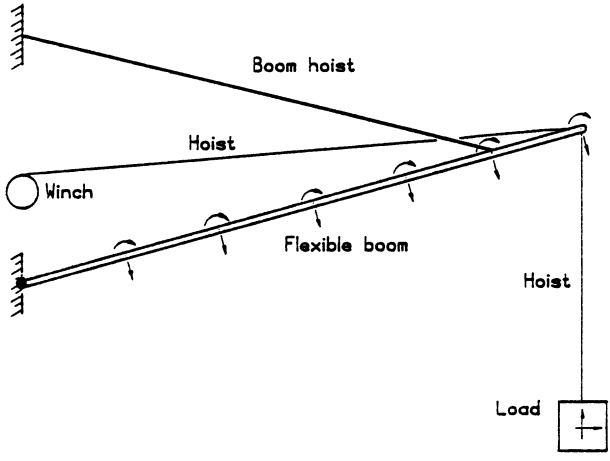


FIGURE 3 MULTI-DEGREE OF FREEDOM MODEL

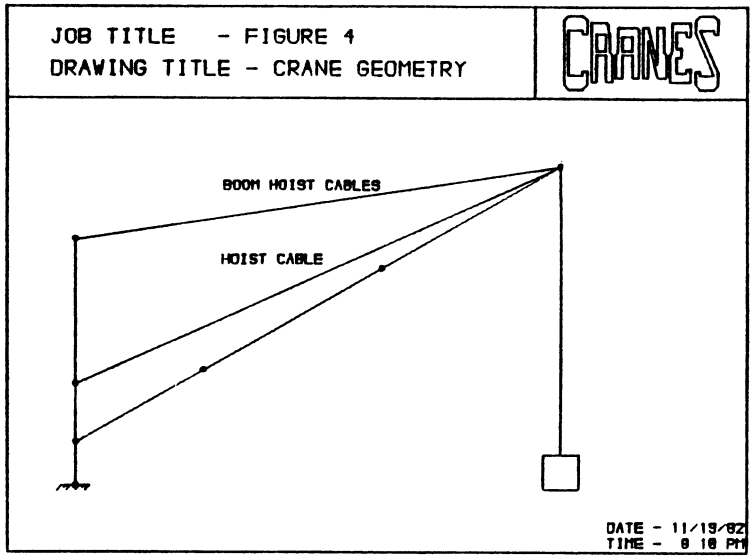


FIGURE 4 SEVEN NODE CRANE IDEALISATION

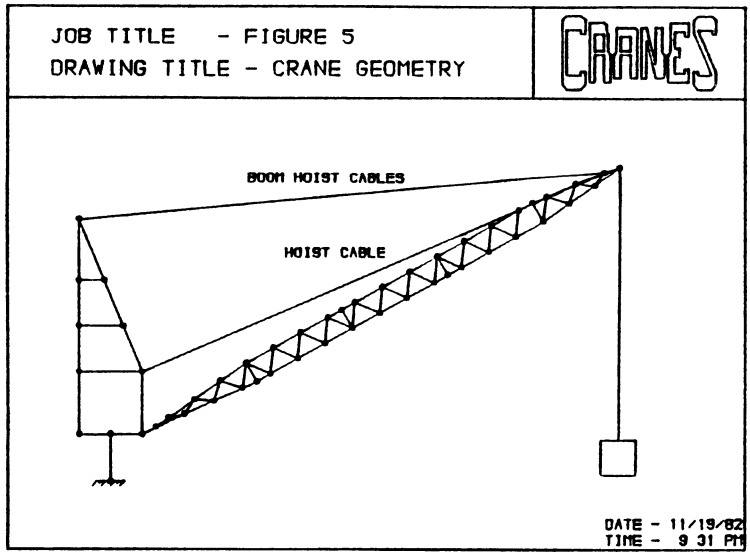


FIGURE 5 FIFTY NODE CRANE IDEALISATION

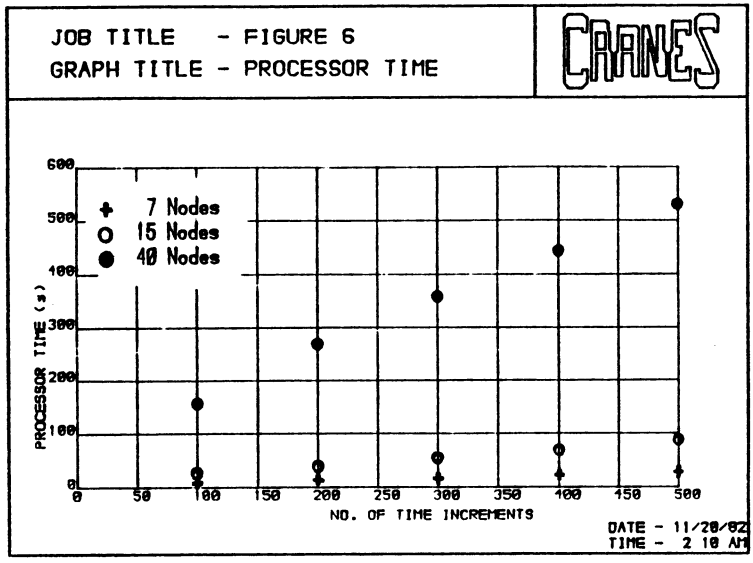


FIGURE 6 DETAILS OF PROCESSOR TIME

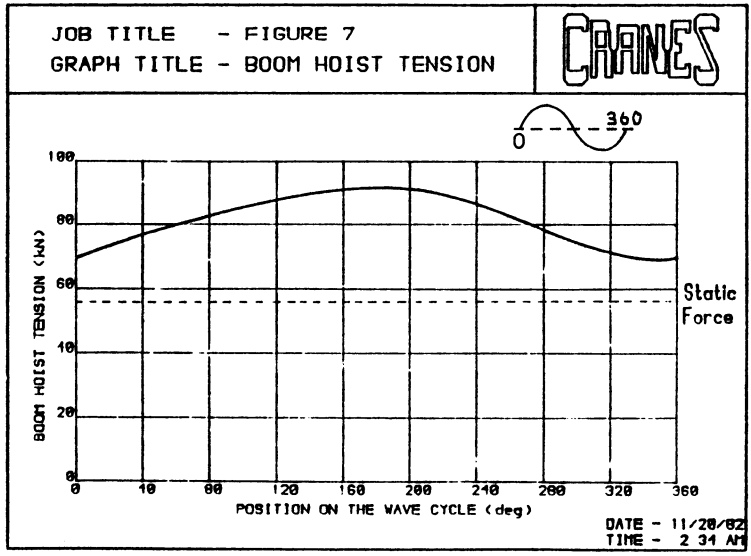


FIGURE 7 BOOM HOIST TENSION vs POSITION OF LIFT-OFF

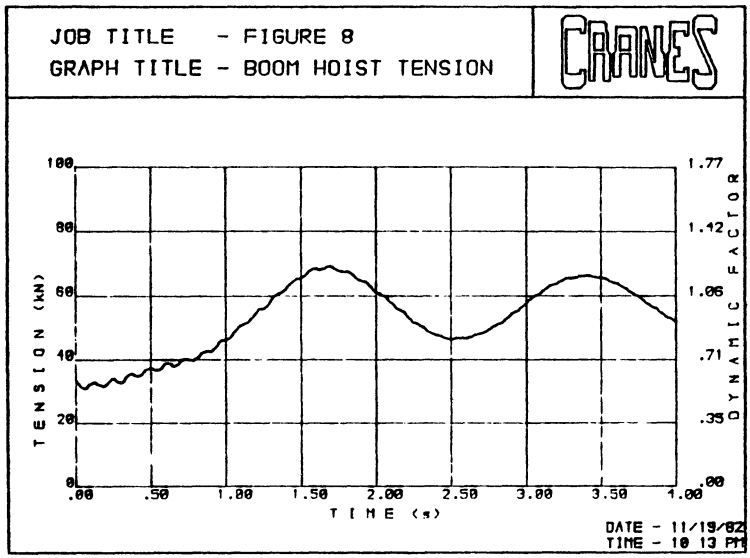


FIGURE 8 TIME HISTORY OF BOOM HOIST TENSION - LIFT-OFF AT Ø DEGREES

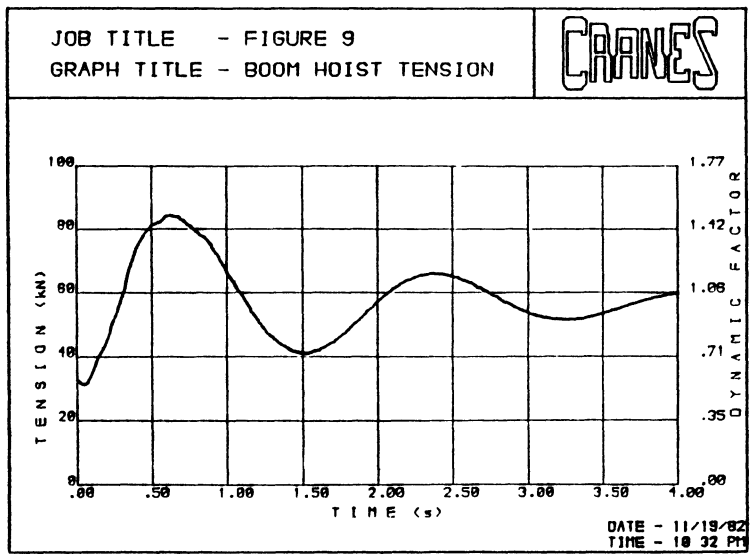


FIGURE 9 TIME HISTORY OF BOOM HOIST TENSION - LIFT-OFF AT 180 DEGREES

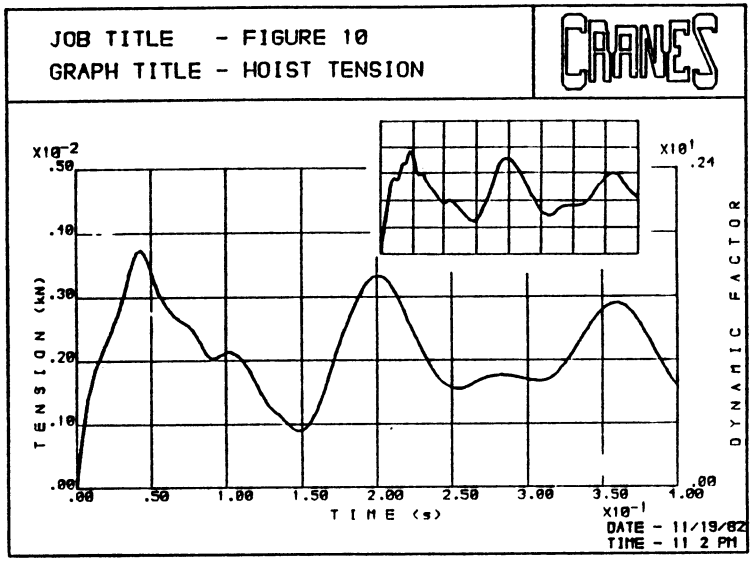


FIGURE 10 HOIST TENSION PREDICTED BY "CRANES" - INSET SHOWS MEASURED RESPONSE

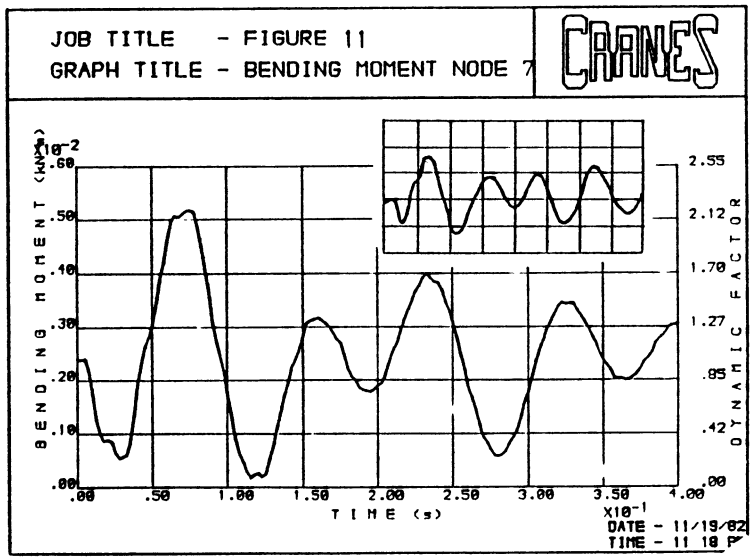


FIGURE 11 MIDSPAN BENDING MOMENT PREDICTED BY "CRANES" - INSET SHOWS MEASURED RESPONSE

APPENDIX 1

```
 CCCCCCC                                                        SSSSSSS
CCCCCCCCC                                                     SSSSSSSSS
CC     CC RRRRRRR                                EEEEEEEE SSS    SSS
CC        RR    RR        A        NN      NN  EE     EE SSS
CC        RR    RR       AAA       NNN     N  EE        SSS
CC        RR    RR      AA AA      NNNN    N  EE        SSS
CC        RR   RR      AA   AA     N NNN   N  EE        SSS
CC        RRRRRRRR    AA     AA    N  NNN  N  EEEEE      SSSSSSS
CC        RRRRRRR     AAAAAAAAA    N   NNN N  EEEEE       SSSSSSS
CC        RR   RR    AA       AA   N    NNN N  EE              SSS
CC        RR    RR   AA       AA   N     NNNN  EE              SSS
CC        RR     RR  AA       AA   N      NNN  EE              SSS
CC     CC RR      RR AAA     AAA   NN      NN  EE     EE SSS    SSS
 CCCCCCCCC RRR     RRR                EEEEEEEE SSS    SSS
  CCCCCCC                                                      SSSSSSS
                    ( RELEASE  3.0 )
```

```
*****************************************************
*                                                   *
*      C R A N E   D Y N A M I C S   P R O G R A M  *
*                                                   *
*    D E V E L O P E D   B Y   J. A. D. B A L F O U R *
*                                                   *
*       H E R I O T - W A T T   U N I V E R S I T Y *
*                                                   *
*                  E D I N B U R G H                *
*                                                   *
*****************************************************
```

```
-------------------------
|                       |
|    D A T A   D E C K   |
|                       |
-------------------------

*TITLE
FIGURE 3
;
*NODES
          1          0.0          0.0
          2          0.0          1.5
          3         -2.0          1.5
          4         -2.0          5.5
          5         -2.0          9.5
          6         -2.0         13.5
          7          0.5          3.5
          8          2.0          5.5
          9          2.0          1.5
         10          3.0          1.5
         11         10.0          1.5
         12         17.0          1.5
         13         24.0          1.5
         14         31.0          1.5
         15         38.0          1.5
C
*ELEMENTS
          1          1          2          1
          2          2          3          1
          3          3          4          1
          4          4          5          1
          5          5          6          1
          6          6          7          1
          7          5          7          1
          8          7          8          1
          9          4          8          1
         10          8          9          1
         11          4          9          1
         12          1          9          1
         13          9         10         -1
         14         10         11          3
         15         11         12          3
         16         12         13          3
         17         13         14          3
         18         14         15          3
C
*BOOM HOIST
    26    100      6      6     15
C
*HOIST
    22    100      1      1      8     15
C
*PROPERTIES
          1          0.0486          0.126
          2          3.392E-3        0.0286
          3          3.008           0.3386
C
*ADDED MASS
         10         500
         11         500
         12         500
         13         500
         14         500
         15         500
;
*CASE
    45    5000     4     1.5     3.5    3     3     1
C
*INTEGRATION AND DAMPING
   0.01       150      0.02     0.02     0.02     0.1
C
*RUN*
```

MATHEMATICAL MODELLING OF THE LOAD BEARING CHARACTERISTICS OF A DYNAMICALLY DRIVEN PILE.

R. Delpak, Principal Lecturer in Civil Engineering
W.M.Hague, Senior Lecturer in Mechanical Engineering
G.C.Lake, Research Student in Civil Engineering
V. Peshkam, Research Student in Civil Engineering
G.O.Rowlands, Principal Lecturer in Civil Engineering,
Polytechnic of Wales, Wales, U.K.

ABSTRACT

The purpose of the present paper is to describe a numerical method for the prediction of the load/displacement characteristics of a partially or fully embedded dynamically driven pile. The system is modelled mathematically so that the soil displays idealised hysterisis in its behaviour while the pile is assumed to be made of relatively rigid elastic segments. Generalised dynamic relationships are applied to the system and a microcomputer is used to provide a numerical solution. Despite the apparently involved representation of the problem the program was kept in its simplest form. A number of examples were examined using the formulation, the results for which are commented upon in the text.

Introduction

Today predictions of static learning capacity from dynamic measurements are mostly based on the one dimensional wave equation first presented by St. Venant, and first considered for use in piling by Isaccs (1931) and Glanville et al (1938). However it was not until the work of Smith (1960) that the method was fully developed. Such workers as Samson, Hirsch and Lowery (1961), and Forehand and Reese (1964) have made considerable refinements to Smith's analysis. A somewhat different approach, involving wave theory is employed by Scanlan and Tomko (1969).Since this time researchers at some American Institutions have made basic contributions to the wave equation analysis. The basic algorithm proposed by Smith has been modified by workers at Texas A and M University to facilitate program usage. The programs produced comprise of TTI (Lowery et al 1968), OCEAN WAVE (Carr et al 1976), and TIDYWAVE(Lowery 1976). The latter program provides for the optional inclusion of load history records as input data.

A rigorous mathematical model of diesel hammers has been developed at the University of Illinois, Rempe (1975), however it's complexity makes its use impractical. The efforts at Case-Western Reserve University have been towards the development of reliable field measurements and sophisticated field interpretation techniques. The first program designated as WEAP (Goble and Rausche, 1976) gives an analysis of the thermodynamics and mechanics of both diesel and air/steam pile driving hammers. A second program CAPWAP employs a wave equation technique which uses measured force and acceleration records to predict soil resistance due to driving, and is currently used by its authors as a viable static bearing capacity predictor.

Also a program DUKFOR (Holloway et al 1975) has been developed at Duke University in which the effects of residual stresses can be incorporated directly in the solution.

The use of the wave equation for predicting the static bearing capacity of soil suffers from two major disadvantages:-

i) A reasonable representation of the hammer system cannot be incorporated.
ii) It cannot be used for piles less than 5m in length.

Theory

In developing the theory for the present analysis the existence of the following principal parameters were recognised:-

a) complicated applied load time history,
b) non-elastic behaviour of the pile shaft,
c) internal damping of the pile,
d) representation of the rheological properties of the soil,
e) an assessment of the mass of soil to be included in the system.

The inclusion of all the other remaining factors would be difficult and so it was decided to incorporate the following simplifying assumptions in the development of the analysis:-

i) The pile was represented in a conventional manner as a lumped parameter system which can be partially or fully embedded, Figure 1. Each section was assigned a unique force/compression behaviour designated by $P_i = K_i \delta_i$. Internal damping of the pile can be incorporated by the inclusion of the term $c_i v_i$.

ii) The resistance to driving is provided by the soil at the corresponding degree of freedom by, a) linear viscous dampers $c_i' v_i$, and b) simulated elasto/plastic behaviour of the soil, see Figure 2. The soil is assumed to have stiffness up to a displacement δ_0 which is termed the 'quake' or maximum elastic deformation of the soil. This may be constant or may vary for each finite depth of soil. Thus if the assumed static resistance of the pile, termed R, is distributed between the embedded segments in any

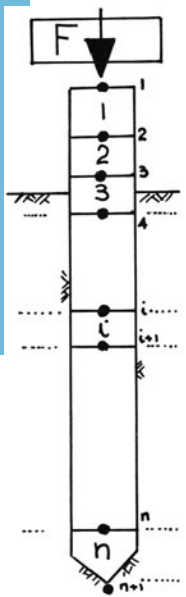


Figure 1.

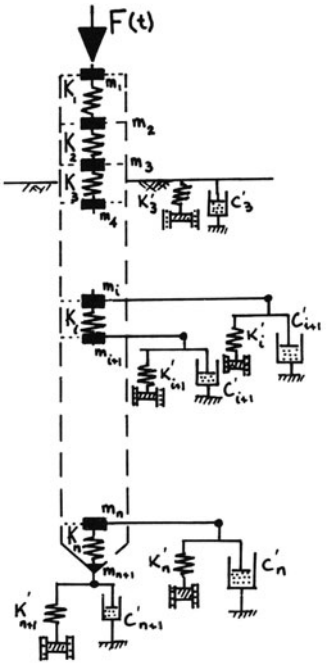


Figure 2.

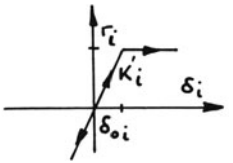


Figure 3.

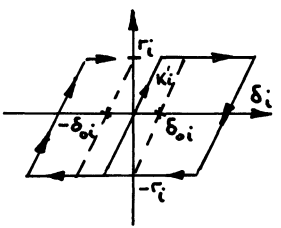


Figure 4.

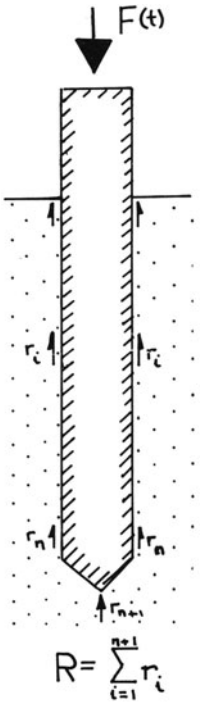


Figure 5.

desired manner, the stiffness for the i th node would be r_i/δ_{oi}, where r_i is the portion of the total static resistance R (Figure 3).

When the 'quake' is exceeded the soil resistance disappears and is replaced at the i th node by the resistance r_i. The exception to this is the last node, i.e. the pile point, which has stiffness $r_n/\delta_{on} + r_n^1/\delta_{on}^1$ and resistance $r_n + r_n^1$, where δ_{on}^1 and r_n^1 are point quake and point resistance respectively. The soil is therefore modelled assuming a two place behaviour pattern depending upon the amount of displacement δ_o. This is similar to the deformation of a slip-joint. (Figures 2 and 4).

iii) The load time history is presented in a numerical form so that for every time increment Δt_j there is a unique value of F_j.

iv) During the elastic range of displacement there is no slip between the pile and soil.

The sequence of calculation in the program is as follows:-

1). The algorithm commences by assembling all the stiffness, mass and damping matrices so that a matrix equation of order n + 1 is set up, i.e.:-

$$[M]\{\ddot{\delta}\} + [c]\{\dot{\delta}\} + [K]\{\delta\} = \{P\} \qquad (1)$$

Soil stiffnesses are calculated as previously described and added to the corresponding diagonal of the stiffness matrix.
The iterations are triggered when the pile is at rest so that:-

$$\{\ddot{\delta}\}_{t=o=o} \quad \{\dot{\delta}\}_{t=o=o} \quad \{\delta\}_{t=o=o}$$

A Newmark method of matrix solution is employed to solve for nodal displacements at each time increment. After each iteration a check is made of displacements, δ_i, to verify that the system remains linear.

2). The process is repeated until δ_o the critical displacement for the node is exceeded, in which case the i th equation is modified to the following form:-

$$m_i\ddot{\delta}_i + c_i\dot{\delta}_i + K_i\delta_i = (P_i - r_i) \qquad (2)$$

The soil spring K_i is now replaced by a constant resisting force r_i as discussed earlier, and depicted in Figure 5.

3). The above continues until all the soil stiffnesses have been replaced by the corresponding soil resistances.

4). At this stage the solution matric, having no constraints, becomes singular and hence insoluble. To overcome this

the pile is considered to become a rigid body penetrating a resisting medium with no internal compresion and hence can be represented by the equation:-

$$M\ddot{\delta} = F(t) - R \qquad (3)$$

5) Equation 3 is only valid for $F(t)>R$ and so when $F(t)$ first becomes less than R, $\ddot{\delta}$ is set to zero along with $\dot{\delta}$ and the multidegree of freedom system is entered with $\{\delta\}_{i=n,n+1}$ equal to the last value obtained from the rigid body equation. This has the effect of redefining the displacement about a new zero point.

6) The next blow of the hammer will be calculated with a new embedded pile length as estimated in steps 1 - 5.

If the calculated and actual displacement curves match then the assumed static resistance gives the static bearing capacity of the pile.

Examples The first example is a simple two degree of freedom system and is used to illustrate that the Newmark iterative procedure is operating correctly. As can be seen from Figure 6 the theoretical and numerical results bear close resemblence. The example is taken from Bathe and Wilson.(1976)

The second example is included to illustrate the various phases of the program. As can be seen in Figure 7 phase one represents the elastic range of the pile where displacements are recoverable, phase two represents the rigid body motion which is non-recoverable and phase three represents exit from Phase 2 and displays some recovery and oscillation about a new equilibrium position.

The third example used the outlined numerical process to compare displacement time curves with that given by Scanlon and Tomko for the same input data. These results are for a pile of the following sizes, area 7.83 sq.in., length 74 ft, and static test bearing capacity of 214 kips. (Figures 8 and 9).

Conclusions The Paper presents an analysis of the problem of dynamically driven piles. It enables the soil parameters and pile characteristics to be modelled mathematically. It is also possible to input the driving characteristics in a numerical form. A specially developed method of solution has been used so that both the linear and non-linear problems associated with pile driving can be investigated.

The results of a number of examples have been verified satisfactorily. The program has been written in Fortran and run on a Superbrain.

Acknowledgements This paper is published by permission of the Heads of Departments of the Civil Engineering and the Mechanical Engineering Departments of The Polytechnic of Wales. G.C.Lake is supported by an S.E.R.C. Studentship.

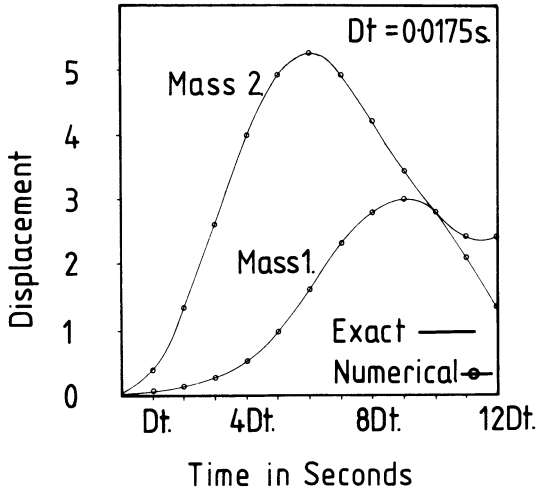


Figure 6.

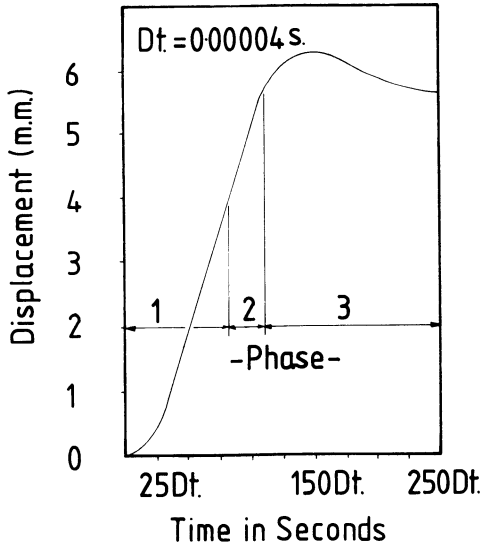


Figure 7.

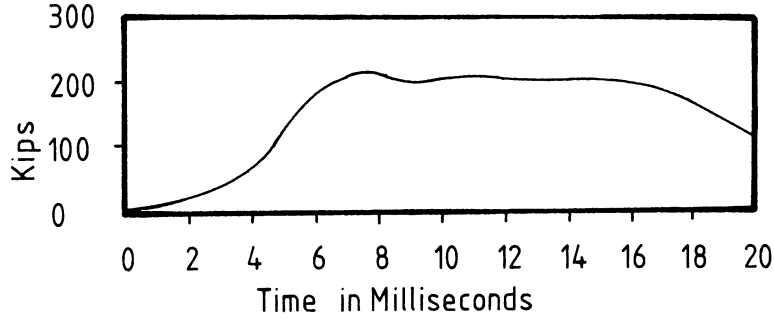


Figure 8.

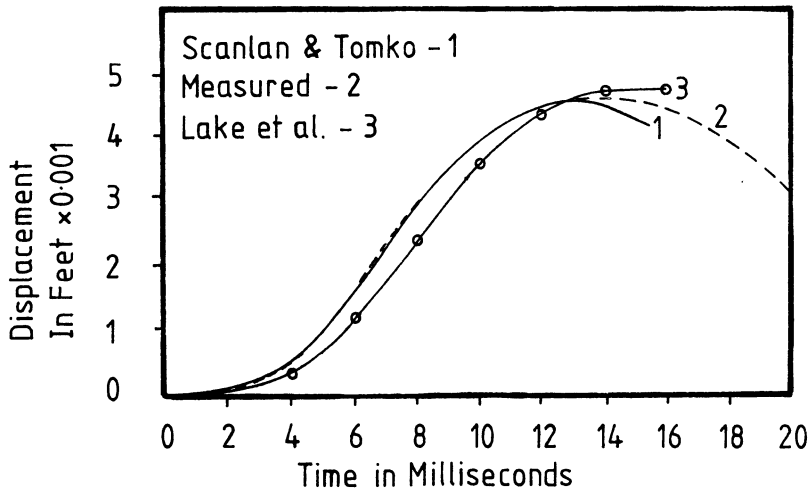


Figure 9.

References

Bathe, K.J., Wilson, E.L., "Numerical Methods in Finite Element Analysis". pp. 332 - 333. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. (1976)

Carr, L., Hirsch,T.J., and Lowery, L.L.Jr., (1976) "Pile Analysis-Wave Equation Computer Program User's Manual (TTI Programme)", (OCEANWAVE), April, 1976.

Forehand, P.W., and Reese, J.L., (1964) "Prediction of Pile Capacity by the Wave Equation". J.S.M. F.D., A.S.C.E., Vol. 90, SM2:1 - 25.

Glanville, W.H., Grime,G., Fox,E.N., & Davies,W.W. (1938) "An Investigation of the Stresses in Reinforced Concrete Piles During Driving". Br. Bldg.,Res.Bd., tech. paper No. 20.D.S.I.R.

Goble, G.G., and Rausche, F., (1976). "Wave Equation Analysis of Pile Driving - WEAP Programme Background - Vol. 1," prepared for the U.S. Department of Transportation, Federal Highway Administration, Implementation Division Office of Research and Development, July 1976.

Holloway, D.M., (1975) "Pile Driving Analysis Using the One-Dimensional Wave Equation", Technical Report S-75-5 U.S. Army Engineer Waterways Experiment Station, Vicksburg, M.S., 1975.

Isaccs D.V. (1931). "Reinforced Concrete Pile Formulae". Trans. Instn. Engrs., Aust., 12:313-323.

Lowery, L.L., Jr., Hirsch, T.J., and Samson, C.H.,Jr., (1968). "Piling Analysis Wave Equation Computer Programme Utilisation Manual", Texas Transportation Institute Research Report 33-11, Texas A & M. University, 1968.

Lowery, L.L., Jr., (1976). "Wave Equation Utilisation Manual", draft users guide for TIDYWAVE, Pile Dynamics, Inc., Bryan, Texas, 1976.

Newmark, N.M., "A Method of Computation for Structural Dynamics" A.S.C.E., Journal of Engineering Mechanics Division, Vol.85, (1959), pp. 67-94.

Rempe, D.M., (1975) "Mechanics of Diesel Pile Driving", Ph.D. Dissertation University of Illinois at Urbana Champaign, 1975.

Samson, C.H., Hirsch, T.J., & Lowery L.L.(1963.) "Computer Study of Dynamic Behaviour of Piling" J. Struct. Divn, A.S.C.E., Vol.89, ST4:413 - 449.

Scanlan, R.H. & Tomko, J.J. (1969). "Dynamic Prediction of Pile Static Bearing Capacity". J.S.M.E.D., A.S.C.E., Vol. 95, SM2:583 - 604.

Smith, E.A.L, (1960) "Impact and Longitudinal Wave Transmission" Transaction, American Society of Mechanical Engineers, 1955.

Timoshenko, S. and Gcodier J.M., Theory of Elasticity, McGraw-Hill, Second Edition 1951, p. 438.

FREEWALL - A MEMBRANE RETAINING WALL DESIGN PACKAGE FOR MICRO-COMPUTERS

L.A. Wood,

Lecturer, Dept. of Civil Engineering, Queen Mary College, London.

SUMMARY

FREEWALL has been developed as a simple to use, truly interactive, software package for the design of sheetpile and diaphragm walls. The program has been written in BASIC and is portable between most 32K + byte micro-computers with a minimum of alteration. Full data editing, disk storage and retrieval of data and results, inspection of results on monitor screen or hard-copy from printer, and many other user desirable facilities have been incorporated in the code for ease of operation.

The comprehensive capabilities of the program are illustrated with respect to several typical wall and soil configurations, encompassing multi-tied walls, sloping and submerged backfill, the development of wall friction and adhesion, and so on.

INTRODUCTION

FREEWALL is a product of an idea to develop a suite of applications programs in the field of geotechnical engineering under the general title of GEOSOFT (Mills and Wood, 1981). The original intention of GEOSOFT was to utilise the opportunities afforded by the screen editting software of the Commodore PET micro-computer, to extend the BASIC interpretater to include plot routines etc, and to provide a single unified database structure for a series of applications programs. However, it was very quickly realised that with the extremely rapid emergence of a wide range of more powerful machines and the "leap-frogging" rivalry between manufactures that a system tailor made to a particular machine would lack universal appeal. Instead it was decided to concentrate on total portability between different micro-computers and different combinations of disk drives and printers etc. To this end,

standard, BASIC has been adopted as the programming language with the absolute minimum of machine dependent code. In this manner conversion from one machine to another requires changes to no more than a few lines of code, which are in any case highlighted for the user both in the source and in the manual (Wood, 1982).

The advantages of a unified database and a single program style are retained and FREEWALL, which is the first, will be followed by a library of applications programs all presenting similar input and output interfaces to the user and accessing unified disk file storage.

A modular approach has been adopted to the structure of each program with complete details of each routine given in the user manual in order to encourage self-maintenance and modification by individual users. In as far as possible the greatest amount of flexibility of application has been incorporated in order to ensure that both future and present design trends may be readily met without recourse to major surgery.

FREEWALL, as the name implies, is a program for use in the design of cantilever, single and multi-tied sheetpile, diaphragm and bored pile walls and cofferdams, exhibiting rotation at the toe. That is a "Free Earth Support" condition is assumed. The operation of the program and the relevant aspects of geotechnical engineering are described below and its flexibility of approach demonstrated with respect to several design problems. Complete results encompassing tie forces, bending moments, shears and earth pressures are computed at each excavation stage.

THEORY

Earth pressures, determined on the basis of the classical Rankine soil mechanics theory of limiting active and passive states, are computed at suitable depth increments down the wall from the pertinent soil and water conditions. It should be emphasised that throughout, all earth pressures are determined on an effective stress basis from consideration of the weight of the soil strata, together with any surcharges, and the pore water pressure, u. Water pressures are assumed to be hydrostatic with respect to the position of the natural water table.

Effective vertical stress, $\sigma_v' = \sigma_v - u$; (1)

where σ_v is the total vertical stress.

Hence, the effective horizontal stress, $\sigma_h' = K_1\sigma_v' + K_2c + q_h$; (2)

where, for the active condition $K_1 = K_a$ and $K_2 = -K_{ac}$;

and, for the passive condition $K_1 = K_p$ and $K_2 = K_{pc}$; these coefficients are calculated from the angle of internal friction, ϕ taking full account of the presence of any wall friction and, or adhesion; or may be user specified: c, is the cohesion intercept and q_h any external horizontal surcharge due to, for example, the presence of adjacent foundations.

If σ_h' becomes negative, indicating tension, a crack is assumed to open, with $\sigma_h' = 0$.

The total horizontal pressure acting on the wall,

$$\sigma_h = \sigma_h'/F + u \; ; \qquad (3)$$

F = 1 for the active condition, F may be user specified for the passive case with $F = F_p$.

The required depth of penetration of wall for rotational stability is computed such that the ratio of the moments of the NET active, M_A and NET passive, M_p forces about the point of rotation are equal to, F_N, a user specified factor. For a cantilever wall the point of rotation is taken as the toe of the wall; for a single tie, about the tie position; and for a multi-tied wall about the line of action of the resultant of all the forces developed in the ties, involving an iterative solution similar to that proposed by James and Jack (1975).

Shears, bending moments and tie forces are calculated for $F_p = 1$ and $F_N = 1$ together with the corresponding depth of penetration. These results are given for each excavation stage in order to determine the "worst case" envelope.

In addition to the specification of factors applied to the passive soil pressure, F_p or the ratio of NET moments, F_N a third method for the provision of a factor of safety is available through the imposition of partial factors on tan ϕ and c. These may be different on either side of the wall. Indeed one of the special features of the program is the provision not only for layered soil deposits but also for different soil conditions on either side of the wall; together with submerged and sloping backfill (angle of slope independent of ϕ), berms in front of the wall and inclined ties or struts.

OPERATION

The program is designed to be totally interactive with the user. Control of the program's action is gained through the use of a number of menus.
The main menu comprises :

N - NEW ANALYSIS/TITLE
S - SOIL PROPERTIES
K - RANKINE COEFFICIENTS
W - WATER CONDITIONS
T - TIE PROPERTIES
Q - SURCHARGES
B - BERMS
I - INCLINATION OF BACKFILL
E - EXCAVATION DEPTH
F - FACTOR OF SAFETY

INPUT ROUTINES

O - OUTPUT MENU
D - DISK FILES (access to database)
C - CALCULATION
R - RESTART (causes initialisation of all variables)

The "D" option enables the user to gain access to the database in order to retrieve, or store, data and computed results. Editting of all variables is continual in that whenever an item of data is to be entered the current value is displayed; this will be left unaltered if "Carriage return" is pressed.

Subsidiary menus control the input of soil properties, Rankine coefficients, factors of safety and output of data and results. The output menu not only includes all of the input routines of the main menu but in addition:

R - RESULTS SUMMARY
P - PRESSURES (calculated earth pressures)
M - MOMENTS AND SHEARS

A - ALL RESULTS (equivalent to RPM)

? - TO PRINTER.

Use of ? causes output from all subsequent requests to be directed to the printer. The results summary provides details of depth of penetration, maximum moments and shears, and tie forces. Note that all screen displays are paginated.

Calculation will not commence until the "C" option is chosen. As the calculation proceeds the user may abandon it by use of a "Carriage return", and return to the main menu. In addition the user may interact with the program in order to shorten the computation time by imparting his or her judgement as to the depth of penetration required and so on. However, this is not always convenient and in order to free the user at this stage, entry of information must commence within five seconds of a request. If not computation continues and all the results for each excavation stage are directed to the printer.

SOIL PROPERTIES

LAYER	TH	UW	C	PHI	CW	DW
1	5	21	5	25	2.5	13
2	6	20	50	0	25	0
3	1000	21	0	34	0	23

TH - Thickness, UW - Unit Weight, CW - Adhesion,

DW - Angle of wall friction.

RANKINE COEFFICIENT

LAYER	KA	KAC	KP	KPC
1	.34	1.43	3.68	4.7
2	.99	2.44	.99	2.44
3	.22	.94	3.69	3.84

DEPTH TO SOIL BEHIND WALL 0

EXCAVATION DEPTH 10
WATER DEPTHS BEHIND 5
IN FRONT 10 OF WALL

MINIMUM FLUID PRESSURE 0

FACTOR ON PASSIVE 1 ON NET 1.5
PARTIAL FACTORS C AND TAN(PHI)
BEHIND 1 1
AND IN FRONT 1 1

DEPTH OF PENETRATION AND RESULTANT SHEAR

FACTOR 1.67 DEPTH 21.5 SHEAR -1521
FACTOR 1.3 DEPTH 20.5 SHEAR -1181.2

FACTOR 1 DEPTH 19.5 SHEAR -880.2
FACTOR .74 DEPTH 18.5 SHEAR -618

FACTOR 1 DEPTH OF PENETRATION 19.5 SHEAR -880.3

MAXIMUM MOMENT 1281.7 AT DEPTH 15
MINIMUM MOMENT -5.6 AT DEPTH 19
MAXIMUM SHEAR 209.8 AT DEPTH 9
MINIMUM SHEAR -880.2 AT DEPTH 19

FIG. 1 CANTILEVER WALL

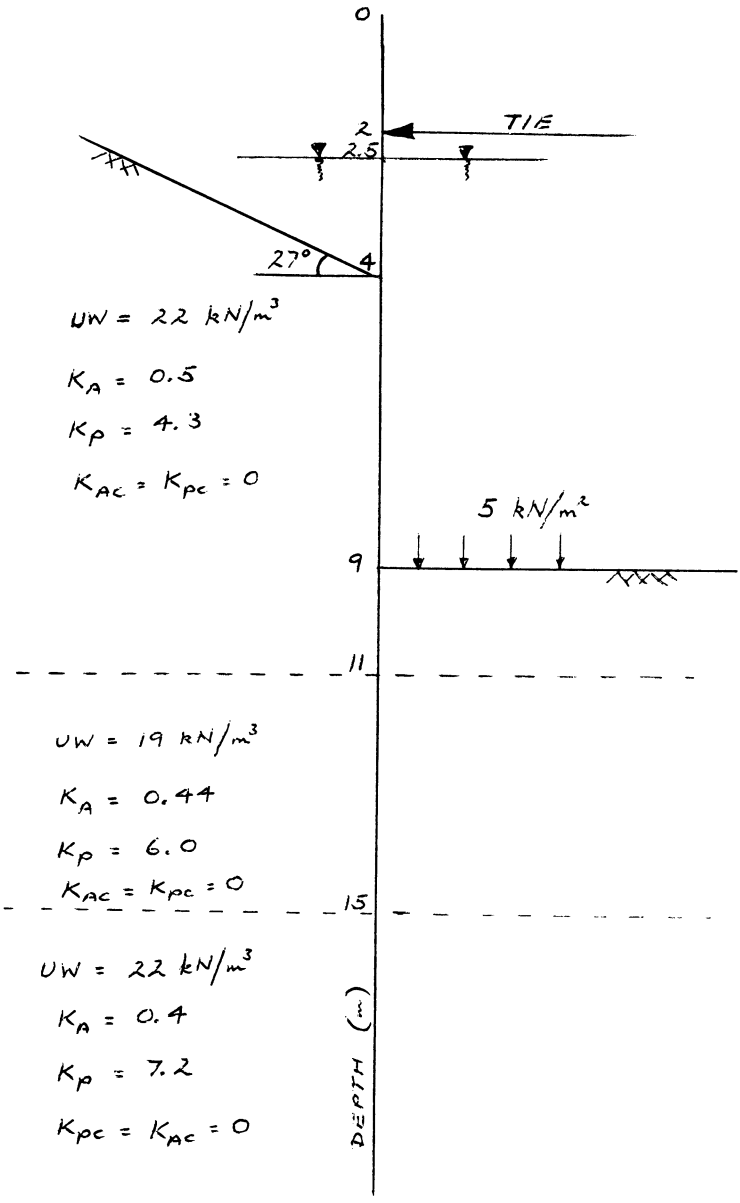


FIG. 2 SOIL CONDITIONS - SINGLE TIED WALL

DEPTH OF PENETRATION AND RESULTANT SHEAR

FACTOR 4.13 DEPTH 12.5 SHEAR -147
FACTOR 1.26 DEPTH 11.5 SHEAR 24

FACTOR 1.26 DEPTH 11.5 SHEAR 24
FACTOR .28 DEPTH 10.5 SHEAR 88.9

FACTOR 1 DEPTH OF PENETRATION 11.2 SHEAR 41.2

DEPTH 2 TIE FORCE 41.2 INCLINED AT 0

EARTH PRESSURES SEQUENCE NO. 2

DEPTH	ACTIVE	PASSIVE	NET
0	0	0	0
1	0	0	0
2	0	0	0
3	4.9	4.9	0
4	14.7	14.7	0
5	33.3	24.5	8.7
6	51.8	34.3	17.5
7	70.4	44.1	26.3
8	89	53.9	35.1
9	107.6	85.2	22.3
10	126.2	147.4	-21.3
11	144.8	209.7	-64.9
12	153.6	324.6	-171

MOMENTS AND SHEARS SEQUENCE NO. 2

DEPTH	MOMENT	SHEAR
0	0	0
1	0	0
2	0	-41.3
3	-41.3	-41.3
4	-82.5	-41.3
5	-123.7	-32.5
6	-156.1	-14.9
7	-171	11.4
8	-159.6	46.5
9	-113	68.9
10	-44.1	47.6
11	3.6	-17.2
12	0	0

FIG. 3 RESULTS - SINGLE TIED WALL

DEPTH OF PENETRATION AND RESULTANT SHEAR

FACTOR 1.79 DEPTH 12.75 SHEAR 126.5
FACTOR 1.12 DEPTH 12.25 SHEAR 192.3

FACTOR 1.12 DEPTH 12.25 SHEAR 192.3
FACTOR .63 DEPTH 11.75 SHEAR 244.3

FACTOR 1 DEPTH OF PENETRATION 12.1 SHEAR 205

DEPTH 1.8 TIE FORCE 71.4 INCLINED AT 0
DEPTH 5 TIE FORCE 133.6 INCLINED AT 0

MAXIMUM MOMENT 58.6 AT DEPTH 12
MINIMUM MOMENT -253.7 AT DEPTH 7.5
MAXIMUM SHEAR 112.1 AT DEPTH 10
MINIMUM SHEAR -109.8 AT DEPTH 5

EARTH PRESSURES SEQUENCE NO. 3

DEPTH	ACTIVE	PASSIVE	NET
0	0	0	0
.5	3.4	0	3.4
1	6.9	0	6.9
1.5	10.3	0	10.3
2	13.8	0	13.8
2.5	17.3	0	17.3
3	20.7	0	20.7
3.5	24.2	0	24.2
4	27.7	0	27.7
4.5	31.1	0	31.1
5	34.6	0	34.6
5.5	38.1	0	38.1
6	41.5	0	41.5
6.5	45	0	45
7	48.5	0	48.5
7.5	51.9	0	51.9
8	55.4	0	55.4
8.5	58.9	0	58.9
9	62.3	0	62.3
9.5	65.8	31.2	34.6
10	69.3	62.4	6.9
10.5	72.7	93.6	-20.9
11	76.2	124.8	-48.6
11.5	79.7	156	-76.3
12	83.1	187.2	-104.1
12.5	86.6	218.4	-131.8

FIG.4 MULTI-TIED WALL

WORKED EXAMPLES

The three examples have been chosen to illustrate the versatility of the program and in order to avoid repetition not all of the results obtained are presented.

Cantilever wall

A summary of the results obtained for a simple cantilever wall supporting a 10m deep cut are shown in Fig. 1 for a factor of safety on the ratio of net moments, $F_N = 1.5$. Also shown are the specified soil properties and computed Rankine Coefficients.

Single tied partially submerged wall

The wall, surcharges and soil properties are shown in Fig. 2. The results obtained for the stage with the tie in place are given in Fig. 3. It should be noted that results for a cantilever wall of the same height were also obtained at the same time (a cantilever stage is assumed to preceed the single tie construction). Comparison of the two sets of results show a reduction in the required depth of penetration from 16m to 12m and a reduction in maximum moment from 400kNm to 170kNm.

Multi-tied wall

The computed penetrations, tie forces and earth pressures acting on a simple 9m high wall supported by two levels of props are shown in Fig. 4. The computed tie forces are:

level one, 1.8m - 71.4 kN/m,
and level tow, 5.0m - 133.6 kN/m.

Depth of penetration required for $F_N = 1.8$ is 12.8m and for $F_N = 1.0$, 12.1m.

CONCLUSIONS

FREEWALL is a truly interactive, user orientated, program. It's modular structure, flexibility of design and the use, in as far as possible, of standard BASIC code will ensure its portability between different micro-computers and its adaptibility to future design and computational trends. Of value in the design office, programs such as this are also of immense value in the education of graduate civil engineers (Wood, 1982). Further application programs, within the field of geotechnics, which utilise the database and routines common to FREEWALL are in the process of development.

ACKNOWLEDGEMENTS

The Apple version of FREEWALL is available from Microsoft, Computational Mechanics Ltd.. The helpful comments and advice of many students and colleagues, particularly Mr. S.Creed,

in the Civil Engineering Department of Queen Mary College are gratefully acknowledged.

REFERENCES

James, E.L. and Jack, B.J. (1975) A design study of diaphragm walls. Diaphrgam Walls and Anchorages. Institution of Civil Engineers, p 41.

Mills, A.B. and Wood, L.A. (1981) GEOSOFT : A geotechnical engineering package for micro-computers. Engineering Software II, ed. R.A. Adey, CML Publications, p 787.

Wood, L.A. (1982a) FREEWALL - User Manual.

Wood, L.A. (1982b) The development and use of micro-software in the teaching of geotechnical engineering. Involving Micros in Education, ed. R. Lewis and E.D. Tagg, North-Holland, p 151.

CIVIL ENGINEERING DESIGN PROGRAMS FOR MICROCOMPUTERS

W.T. Bell and R.J. Plank

Department of Civil and Structural Engineering
University of Sheffield

INTRODUCTION

The use of the new generation of microcomputers in civil and structural engineering is now well established and this paper sets out to review the various types of applications, their relative successes and drawbacks, and seeks to draw some basic conclusion on the main areas requiring development in the future. The traditional use of computers, and the dangers of extending traditional computing techniques to microcomputers have already been discussed (Bell, 1981), but developments over the last two years warrant a review of how microcomputers have been applied.

MICROCOMPUTERS IN ACTION

Since the first appearance of the cheap mass-produced microcomputers in 1979 there has been a variety of firms offering computer-aided design programs, varying from the simple and highly specific programs to the 'all-singing, all-dancing' software packages. With the benefit of hindsight it is easy to criticise the early attempts to integrate the microcomputer in the design process, but a review is worthwhile to show how ideas have changed over a relatively short period of time.

Microcomputer applications have progressed through at least three stages: the downgrading of existing mainframe programs with little attempt to utilise the full interactive facilities of microcomputers; the re-writing of downgraded mainframe programs to allow interactive data input; and the development of completely new microcomputer programs. These applications will be discussed in detail.

DOWNGRADED MAINFRAME SOFTWARE

The easiest and quickest way of obtaining a microcomputer program was to re-write existing mainframe programs. The task was relatively easy because although mainframe programs were usually written in FORTRAN, the conversion to BASIC with its very similar program structure was a simple matter of re-coding and keying the program into the micro. However, mainframe programs use data provided either on punched cards or in an on-line filestore which requires the program to read data in a serial format and to process and produce results on a 'data-in, results-out' basis. Some of the early writers of micro programs followed exactly the same mainframe program structure, even to the extent of data having to be included in the source program as DATA statements used in BASIC. While this approach obviously utilised cheap computing power in the absence of other, more appropriate, programs, it is evident that virtually no use was being made of the interactive facilities available on micros.

INTERACTIVE DATA PREPARATION

The next stage of development was to convert the main processing routines of mainframe programs, but to write interactive data preparation programs to enter checked data into a filestore or direct to the processing program. This overcame the problem with conventional 'data-in, results-out' approach where minor errors of syntax or logic in the data would not appear until the processing program had actually tried to run. Although interactive data preparation undoubtedly reduces the time lost in data errors, a complete new set of data had to be entered to run the program again with only minor changes in the data. There was only limited ability to explore the sensitivity of the results of a design to the input data.

Although few of the interactive facilities of the micro are used in the main processing routine, it should be recognised that there are still a number of applications which are fundamentally unsuited to interactive processing. For example, the standard stiffness method of plane frame analysis is unable to accommodate the engineer's expertise part way through the analysis; the results arise only at the end of the processing routine and only then is the engineer able to use his experience to judge the efficiency of his structure.

MORE RECENT MICROCOMPUTER SOFTWARE

A considerable amount of effort and money has been invested recently in 'the new approach' previously advocated (Bell, 1981). Many design problems in practice are solved quite adequately without the use of a computer because the engineer

uses his judgement and experience to arrive at a suitable design. The greatest danger of the computer is that it may tend to usurp the engineer's role in design; while we can wonder at the kilobytes of memory contained in a cheap micro, we should acknowledge the gigabytes of memory which the engineer has (Burgess, 1980) and let each do what they are best at - the engineer to think and the computer to carry out the mundane and error-prone calculations.

The more recent application software has tried to build on this principle by allowing the engineer to react appropriately as and when information is produced by the micro. The best example of this is in reinforced concrete design where a certain amount of design has to be carried out before the engineer knows that his ideas are feasible, and where the requirements, for example, of a practical bar layout may dictate the best solution. Although attempts have been made to automate the design process from structural geometry through to a bar bending schedule, it is difficult to accept that the computer is sufficiently versatile to replace engineering intuition when practical requirements dominate the theoretical structural requirements.

The principle has been adopted that the engineer's interaction with the computer is no more than his interaction with an electronic calculator, and we now have the basis for developing 'appropriate' programs for micros. An engineer uses his calculator as a convenience: it not a pre-requiste of design and much of his time should be taken upon with 'calm appraisal' (Ryder, 1980) of the problem. The micro is now viewed as being an extension of the normal long-hand design process.

RECENT TRENDS

Although micro software has developed in the generally required direction, there are now disturbing indications of a trend back to the 'all-singing, all-dancing' approach taken in mainframe programs. This is a result of two features; first, the increasing level of familiarisation of engineers with microcomputers and the confidence that has developed over the last three years, and second the incredible increase in computing power (with respect to cost) of the more recent 16 bit micros. These two factors have combined with the result that we are able to do yet more and more on a relatively inexpensive computer system. We believe that the memory restriction on the 8 bit micros of 64K (say 30K of program and data space) provided an excellent way of controlling the size and complexity of micro programs. The danger now is that with 512K available on 16 bit systems, there will be attempts to produce programs to suit every eventuality.

The feedback from industry resulting from courses on microcomputer applications at Sheffield University is somewhat mixed on the question of what level of complexity of software is needed. There is often a difference of initial view between the private and public sector engineers on the question of the degree of automation required in their work. Although there have been notable dissentors, the first reaction from the private sector is that they would be quite happy with the idea, if it could be done, of entering basic data at the start of a problem and allowing a computer to produce everything from design calculations to drawings and bar schedules. However, on probing for the reasons for this view, it usually transpires that it is not the design process itself which is thought to be arduous, but the difficulty of finding good detailers and draughtsmen to carry out the relatively mundane tasks of producing the final product of a design office. We suggest that the choice of the most appropriate design method should not be a consequence of the need for a computer draughting system; the two should be considered independently.

The other main differences between the private and public sector on the question of software needs is concerned with the question of the source of programs. The private sector, claiming that they are more cost-conscious than their public sector colleagues, will argue that it is more cost-effective to purchase off-the-shelf software, although it means adopting a design technique implanted by the program author. The public sector, however, are considerably more willing to develop their own programs to meet a particular need and using existing manual techniques. While there is the danger that there will be numerous engineers all over the country each re-inventing a continuous beam analysis program, the Design Office Consortium (now the Construction Industry Computing Association) believe that "the need for the existence of programs with a variety of approaches to structural design is greater than the need to avoid duplication" (DOC, 1980). We endorse this sentiment if only to avoid engineers surrendering their own professional expertise to somebody else's interpretation of design contained in a program.

FUTURE DEVELOPMENTS

The second part of this paper will further develop our ideas for 'appropriate' micro software, along the general lines of 'utility' or 'tool-kit' programs. However, it should be stressed that the micro still has a role to play, although rather specific, in larger applications involving the pre-processing of data ultimately destined for mainframe processing. There are several applications where even 16 bit micros would have difficulty in design (except at an initial feasibility stage). For example the BIPS and HOPS suites for

highway design are widely available on mainframes and it would be inappropriate to attempt to replace the vast investment in these program suites to transfer them onto a micro. Indeed, there appears to be no reason to do so because a micro can be used as an intelligent terminal to a mainframe to prepare and transmit input data and to receive the results. This particular application is a good example of our previously discussed fears of the effect of more powerful micros, because although 8 bit micros are used to explore the feasibility of initial highway alignments, it would be unnecessary to replicate BIPS on a 16 bit micro.

THE NEED FOR 'APPROPRIATE' MICROCOMPUTER SOFTWARE

It is apparent that even in firms with significant computing facilities a great deal of work which could effectively be done with the aid of a computer is still being done by hand. The most common type of application of computer aided design is in employing 'more accurate' methods to the solution of a particular problem - for instance, using a plane frame analysis incorporating the complete structural framework rather than a series of anlaysis by hand of limited subframes. In theory this may be desirable but the accuracy of any results will depend not only on the sophistication of the analytical method employed, but also on such things as the accuracy of the input data. Using again the example of a plane frame analysis, items such as loads and stiffness involve approximations to some degree, and these implicit 'inaccuracies' may be far more significant than those associated with simplified solution techniques. Nevertheless the profession still seems drawn towards this type of application - perhaps because it is the type of application to which computers have traditionally been put - and there is relatively little use of computers in the day-to-day work of the engineer, where manual methods of design are regarded as adequate. However it is quite clear that if the computer were used in many of these routine tasks, considerable savings in the engineer's time could be realised. Furthermore, it would be the tedious, mundane, repetitive tasks which the computer would be undertaking, allowing the engineer to devote far more of his time to the more important aspects of concept design and the exploration of alternative solutions.

Such applications are of course dependent on the availability of suitable software. In general the requirements for such software are firstly that it will serve the purpose, and secondly that it is easy to use. The former may seem to be perfectly obvious, but it is perhaps worth underlining that if an engineer wants to design a simply-supported steel beam (not an uncommon task in a structural design office) he does not need a program which is also capable of designing reinforced concrete beams, steel

beams in a rigid frame or steel beam-columns. What he requires is a simple steel beam design program - a simple program dedicated to that particular application - because this will help to achieve the second requirement of ease of use. Furthermore because the scope of such programs would be clearly limited, and the content therefore relatively simple, it becomes quite feasible for the engineer to write his own programs. This offers further advantages in that he can follow exactly the procedures which he has traditionally used, and he should be fully aware of the capabilities and limitations of that particular program.

The example of the design of a simply supported steel beam will now be used as an illustration of this type of application. The procedures to be followed are briefly set out below, together with documents to which the designer may need to refer

	Task	Reference
	Calculate loadings, bending moments etc.	Reference book - e.g. "Steel Designers' Manual" (Constrado, 1972)
→	Select trial beam section	Section tables (BCSA/Constrado, 1978)
	Calculate design parameters	BS449 (BSI, 1969) "Steel Designers' Manual" (Constrado, 1972)
	Compare with permissible values	BS449 (BSI, 1969)
NO —	Is the section satisfactory?	
	YES	
	↓	
	END	

It is apparent that even for such a simple task the engineer may need, at various stages of the design process, to refer to a number of documents or carry out some simple calculations. By providing the required information within the program and incorporating the necessary calculations, allowing the engineer simply to supply the data, select the trial section sizes and decide - on the basis of the results

calculated - whether or not the section is satisfactory, all the tedious work is carried out by the computer. If this represents the 'rough working' to arrive at a suitable solution, presentation calculations can also be produced by incorporating a routine to print the calculations for the selected section in the required detail on the printer.

It is perhaps worth drawing attention to a number of points highlighted by the above example.

(a) The scope of the problem is very limited, the problem itself is relatively simple, and consequently the program is not a vast complicated piece of software but is simple enough to be written by the the engineer himself.

(b) The calculations and procedures adopted are identical to the traditional manual methods.

(c) None of the 'engineering' is done by the computer, which is acting simply as a reference manual and calculator - not a decision maker. (No doubt some would suggest extending this type of program to include some degree of 'optimisation', but whilst it might seem attractive to automatically derive the 'best' solution it is surely the engineer who is in the best position to decide what constitutes the 'optimum'.)

(d) Although the application and the program itself are relatively simple, the potential benefits are not inconsiderable.

The example used is that of a steel beam design, but there are many other similar tasks to which the approach described above could be applied. Indeed almost all routine engineering design work would be suitable for this type of simple application. Even the apparently trivial example of determining loads on a structure or on a particular structural element could be usefully undertaken on a microcomputer. This task done by hand is tedious, time consuming, and is certainly not one which gives the designer any sense of achievement. With the increasing use of limit state design and partial safety factors for loads which will vary for different types of loading and design conditions, the calculation effort is multiplied and hence the value of computer application is increased.

There has been a tendency to link microcomputer programs together - thus in the above example the loads calculated would subsequently be used in the design of slabs, beams, columns and foundations with data carried forward from one set of calculations to the next. Whilst the attraction of this may be fairly apparent, such a treatment inevitably limits the flexibility of individual parts - e.g. that

dealing only with beam design forces the user to adopt the procedures used in the program in its entirety and not least contributes greatly to the complexity (and cost) of the programs themselves. (It would not be unreasonable to estimate that a doubling of the scope of the program would - including error trapping, facilities to go to individual parts of the program etc. - increase the total program complexity by perhaps a factor of 4. This leads back to the familiar problem that complex programs are best written by programmers, yet it is the engineer who is the most suitable person to write programs for an engineering application. By limiting the aims of individual programs it becomes reasonable for any engineer with a little interest in computing to develop a set of programs for his particular application. Clearly in one firm or office it would not be sensible for every engineer to develop his own programs thus duplicating effort and software, but in many cases there are office procedures - methods of design which, whether stated formally or not, would be followed by all individual engineers in the office - and clearly it would be appropriate for one person (in consultation with others) to develop a program for use by the whole office. For simple programs such as those outlined above the effort should not be great.

THE DRAWBACKS OF AUTOMATED DESIGN PROGRAMS

Reference has already been made to the question of optimising routines within design programs. The apparent advantages of optimisation are in firstly overcoming the inherent drawbacks of batch processing, and secondly providing the most 'economic' solution. The latter would seem to be self evident until the means of achieving an optimum solution is studied. More likely than not the program author and the user will have different ideas about the criteria for 'economy' and in these circumstances the 'optimising' advantage in fact becomes a disadvantage. In many cases for instance the simple criterion of minimum weight is incorporated within a design program, but in many instances, availability of stock, or detail considerations may be more important. It is therefore argued that the advantages of optimising are outweighted by the disadvantages, and that in any case it would lead to an unnecessary complication of the program. Furthermore, whereas big automated 'optimising' programs require little of the engineer other than data input (and the decision to use that particular program) thereby rightly raising fears within the profession that there would develop a generation of young engineers who had no sensitivity or feeling for engineering solutions, the approach advocated in this paper not only avoids this but in fact provides a basis on which young engineers might very readily gain 'experience' of sensible solutions. Provided the software is good, data will be capable of being edited and hence the inexperienced engineer can explore the

sensitivity of his design to various parameters. This will enable him to develop greater feeling for those parameters which are important and those to which the final solution is relatively insensitive. Such a luxury is rarely available today to someone using manual methods of design.

The second advantage of 'optimising' programs - that of overcoming the inherent drawbacks of batch processing - is something which is negated by the very nature of the microcomputer itself. Of course the programs referred to are often large and even if it were possible to mount them on micros they would probably be slow to execute compared with the speed of execution on a mainframe. However much of the size and time are a direct result of the in-built optimising process. If one recognises that the users of microcomputers in design offices are qualified engineers seeking to use the computer more in the way they now use calulators, rather than someone with no knowledge of the solution techniques embodied in the program being used, it is clear that the system - consisting of the computer (hardware and software) and user - is considerably more powerful than the nominal 64K of the machine. Indeed the ability of the human mind to think creatively, use engineering judgement or make decisions based upon feel rather than precise logic, is something which even the largest computer could not equal. It thus makes good sense to utilise this power to complement the more pedestrian (albeit quick) ability of the computer. Microcomputer programs should therefore be written with this in mind, with the programs being kept relatively simple and full use being made of the user's own special ability. Citing again the example of the steel beam design, one item required in the calculations is the effective length of the compression flange. Whilst given particular conditions it is possible to determine an appropriate value for this parameter, this would require a relatively complicated section of program to cope with all possible cases, whereas if the user is simply asked to decide himself the effective length (or the factor by which the span should be multiplied to obtain this) this part of the program becomes trivial. If necessary it might be appropriate to provide a display reminding the user of various typical cases and corresponding effective length factors. Apart from making sensible use of the user's abilities, this approach would relieve the user of considerable amounts of data input.

CONCLUSIONS

This paper has presented a brief review of the developments in software for microcomputers and highlighted some of the pitfalls which have become apparent. A philosophy for the development of new microcomputer software has been given with a case being made for simple utility programs. The need to

involve the user of the computer fully as an integral part of the system rather than strive for optimising or automatic design programs has been emphasised.

REFERENCES

Bell, W.T. (1981) The Role of Microcomputers in Civil and Structural Engineering. Engineering Software II (Ed. R.A. Adey), CML Publications, Southampton.

Burgess, I.W. (1982) Programming Engineering Applications Microcomputers in Civil and Structural Engineering. University of Sheffield.

Ryder, T. (1980) The Engineers Computer Handbook, Thomas Telford Ltd., London.

D.O.C. (1980) Micros in Construction. Design Office Consortium, Oxford.

Constrado (1972) Steel Designers' Manual, Crosby Lockwood Staples, London.

BCSA/Constrado (1978) Structural Steelwork Handbook.

BSI (1969) BS449 : The Use of Structural Steel in Building.

SOLVING CONCRETE CRACKING PROBLEMS WITH A GENERAL PURPOSE FINITE ELEMENT PROGRAM

Erik Grodtkjær

Concrete and Structural Research Institute, Lyngby, Denmark

INTRODUCTION

FIESTA (FInite Elements in STructural Analysis) is the name of a general purpose computer program for the analysis of structures with linear or non-linear material properties. FIESTA has been developed during the last few years at the Concrete and Structural Research Institute. Originally the program was intended only for nonlinear and cracking analysis of concrete shells of revolution with arbitrary loading, using a semi-analytic method in which the circumferential variation of the load is approximated by Fourier series. This method is widely used in linear analysis, but very little in nonlinear analysis. Since a program with this capability has to possess most of the necessary attributes of a general purpose program, it was decided to continue the development with appropiate regard to generality.

Other important aspects of the philosophy, on which the design of FIESTA is based, are:

1. The program must be user-friendly. To achieve this, FIESTA is supplied with free-format input, extensive data generation and error checking facilities. The manual is kept at moderate size, but is concise and detailed enough for the beginner.
2. The elements should be robust and versatile. Some large FEM systems support as many as 100 element types, many of which seem quite similar, and the user is often forced to select an appropiate element for his problem without knowing which type is best suited. FIESTA has only a few element types, but each type is general enough to model any possible structural design within the relevant type.
3. The program should be independent of hardware and operating systems to obtain portability.

FACILITIES OF FIESTA

Input

Input is given in a number free format data groups. These are described in detail in the manual (Bergholdt and Grodtkjær 1981) with an easily learned syntax, and the user will soon be able to write the input by merely referencing syntax charts for each group. Most groups contain data generators. In the data groups for nodal coordinates and element topology, generation can be carried out in more than one dimension. The generated element net can be plotted for quick error-checking.

```
*NODE
< SPACE
  PLANE
  ROT
  CYLINDER
  SPHERE
{node1  x1 [y1] [z1]
 (line1) [GENERATE n-lines [SPACING fac3 fac4] (line2)]}
>
 line:
{node1  x1 [y1] [z1]
   GENERATE incr [SPACING fac] node2 x2 [y2] [z2]
 CURVE n-nodes <node-num x [y] [z]>}
```

Table 1. Example of data group syntax chart.

Element types

At present FIESTA contains 5 element types, which are listed below along with their main characteristics. The number in the element name indicates the number of degrees of freedom:

RS12	axisymmetric curved thin shell element, 2 nodes, non-axisymmetric load
RB4	ring beam element with stiff cross section, 1 node, non-axisymmetric load
M12	triangular membrane element, 6 nodes
P12	triangular plate bending element, 6 nodes
S24	triangular flat shell element, 6 nodes
RM18	axisymmetric, triangular element, 6 nodes

All elements (except RM18) may have varying thickness and may be composed of different material layers.

Materials

Each material may be isotropic, orthotropic or anisotropic,

and have nonlinear properties (yielding, cracking etc). Any nonlinear material model can be inserted. At present, however, only models for concrete and reinforcement are included.

Loading

The loading on each element may consist of point-, line-, surface-, volume-, and temperature loads, prestressing, initial strains and stresses. Loading may be described in local or global coordinate systems. In the axisymmetric elements the arbitrary variation of the loading in the circumferential direction is described by Fourier series, which may be generated automatically by the program.

Supports

Support directions may be given in local, global or rotated (for skew supports) coordinate systems. The supports can be fixed or elastic, and displacements can be described.

Special structural properties

Structural hinges can be modelled in any element type, and rigidly connected nodes be specified.

Type of analysis

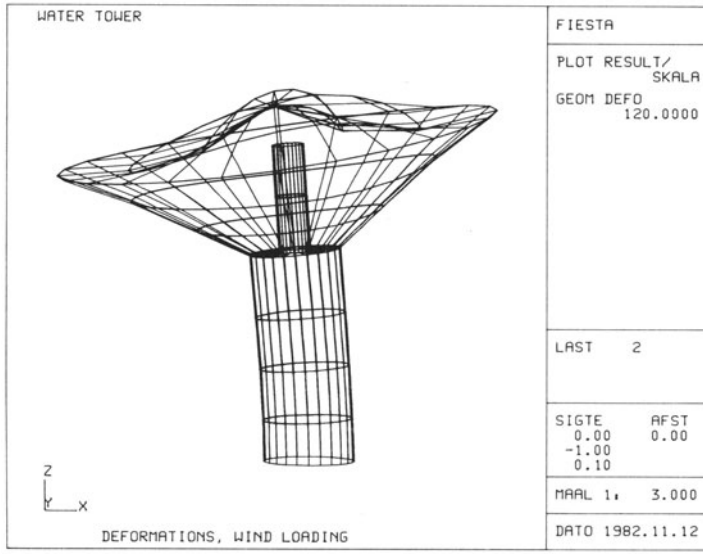
Static, linear and nonlinear analysis can be performed. In linear analysis an arbitrary number of load cases and loading combinations can be treated. The nonlinear analysis is used in problems with nonlinear material properties (including cracking). The loading, which is applied in steps, may be a combination of different load cases. The factors of these on each step can be different so that, for example, one load case may be the dead load. Near the ultimate load the step size can be reduced automatically.

Results

Nodal displacements as well as support reactions are available. In the elements section forces and/or stresses at the top and bottom of each material layer can be calculated. Extensive facilities exist for the selection of results to be printed or plotted. Plots of the undeformed and deformed structure, contour plots and diagrams of any result, vector plots of principal stresses or forces/moments can be produced on any graphic device. The plotting is carried out most conveniently by simple, interactive comm nds. All necessary graphic routines are included in FIESTA, making it independent of standard graphics packages. Figure 1 shows an example of a plot and the command used to produce it.

Method of solution

The equation solver is a Cholesky factorization algorithm (without square-root calculation making it more efficient than



Command:

```
GEOM DEFO 120  ALL  LOAD 2  VIEW 0 -1 .1
LABEL TEXT DEFORMATIONS, WIND LOADING/ PLOT CAB1
```

Figur 1. Example of plot with command

the normal Cholesky algorithm). The non-linear analysis utilizes the modified Newton-Raphson method in which the stiffness matrix is kept constant (Grodtkjær and Bergholdt 1982).

Size of problems

FIESTA contains no limitations on problem size - only the computer hardware sets the limit. FIESTA is supplied with semi-dynamic storage allocation so that the necessary core storage can be adjusted to the actual problem. If the core storage is insufficient, secondary storage is used automatically.

Availability and implementation

The source program of FIESTA is sold in modules so that the costumer does not have to buy parts of the program which are of no interest to him. The prices are low compared to other FEM systems.

Until now FIESTA has been implemented on the following computers: IBM (S370, 43xx, 30xx), PRIME (400,500), DIGITAL (VAX, DEC), UNIVAC (1100), CDC (CYBER 723). Written in standard FORTRAN it can run most mini-computers or main-frames.

CONCRETE CRACKING

One important application of FIESTA is the nonlinear analysis of unreinforced and reinforced concrete structures in which cracking may occur. Of special interest is the use of the semi-analytical method in axisymmetric structures with arbitrary load. It was found that this method is economically superior to a fully three-dimensional analysis. The application of the method to cracking problems is demonstrated in the following two examples.

Pinched cylinder of unreinforced concrete

This example shows the capability of the method to follow the stress redistribution occuring as cracks develop in the concrete when the loading is increased. The structure (see figure 2) is an idealized sewer pipe in an idealized test loading situation (the pipe is not supported).

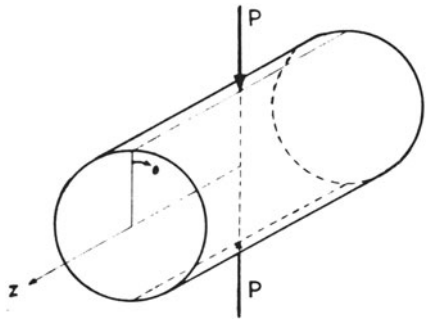


Figure 2. Pinched cylinder

The structure is modelled with 8 axisymmetric RS12 elements along half of the cylinder (see figure 3). The lengths of the elements decrease towards the point forces. These are modelled using a Fourier series of order 15. The basic point forces have the value 2050 N, and are applied in 5 steps with the factors 10,14,18,22,30, respectively.

The stress-strain curve used for the concrete is shown on figure 4. Since the compressive stress will never reach a very high level, the curve is assumed linear in compression. The concrete cracks when the tensile strain reaches 0.0002.

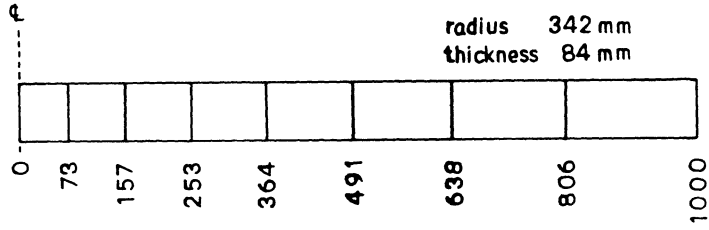


Figure 3. Element model

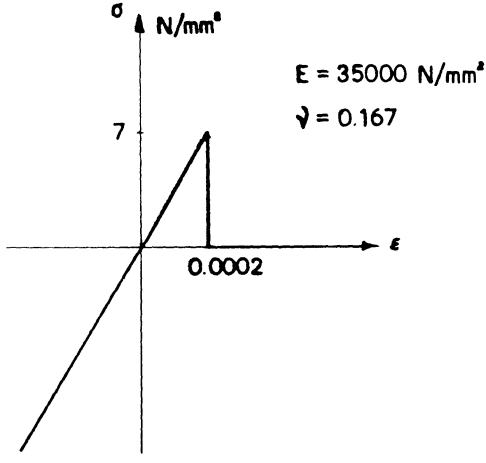


Figure 4. Stress-strain curve for concrete

Table 2 shows the complete input to FIESTA for this problem. The stress-strain curve is defined by 4 points. The Fourier series approximation of the point forces is obtained conveniently through data group *FOURIER.

```
*TITLE
PINCHED CYLINDER, UNREINFORCED CONCRETE, RS12
*NODE RUT
 1 342 0  GEN1  3P 1.15   9  342 1000
*ELGEOM
 RS12 1 90 84 90 84
*MATERIAL
  1 3.5E4 0.1667
  MODEL 1 P -1 -3.5E4 0 0 .2E-3 7 .2E-3 0 1 0/
*ELEMENT 1 RS12
 1  1 2 1 1  GEN1 8  A1  1 1
*SUPP 1 3F 5F / (SYMMETRY COND)
*LOAD 1
 RS12 LINE 1 EQU1 LOCAL 1   0 -6
*FOURIER 1 15 1
 DELTA 13 2  DIRAC 90 1  DIRAC 270 1  SETC A 1 0
*OUTPUT
DIRS PLOT ALL T EQUI 36/
 STEP 1 2 3 4 5 6 7 8 9 /
*ANALYS STATIC NONL 9  COND 50 0 1E-4 5E-2
              INDIV 10 14 18 22 26 30 32 34 36  1 /
END
```

Table 2. Input to FIESTA

As the loading is increased cracks develop inside the cylinder at the point forces and then propagate in axial direction. This will cause a redistribution of stresses until cracks develop at the sides (on the outside). The collapse load has then been reached.

Figure 5 shows the hoop stress σ_θ along the inside top of the cylinder for different load levels. On the first level the cracking strain has just been reached. On the following levels it is seen that the stress drops to zero when the cracks is formed, and that it propagates to the right.

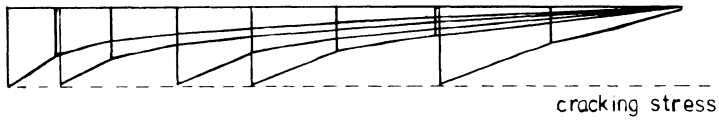


Figure 5. Inside hoop stress, $\theta = 0$

Figure 6 shows, for the same load levels, the variation of σ_θ around the outside of the cylinder in the plane containing the point forces. The last load level is just before the cracking stress is reached at the sides.

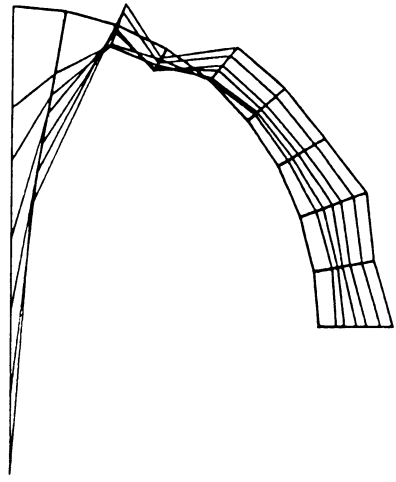


Figure 6. Outside hoop stresses, z = 0

Stress redistribution in concrete storage tank

In this example the method is used to analyse a reinforced concrete storage tank as shown in figure 7. In such tanks the design criterion for the reinforcement is often the avoidance of excessive cracking in order to keep the fluid from seeping through the walls.

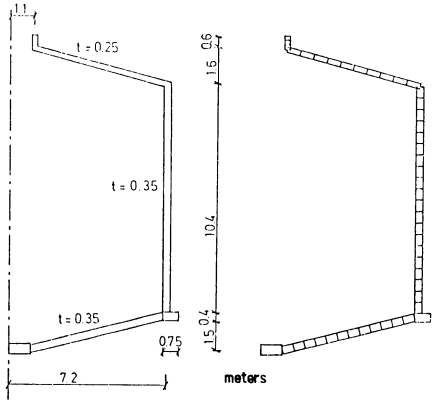


Figure 7. Storage tank and element model

The structure is analysed for an axisymmetric load case (the liquid in the tank). The linear and nonlinear response of the tank are calculated, the latter by taking into consideration the development of cracks in the concrete. This causes a change in the stiffness of the structure and thereby a redistribution of the stresses. This redistribution can be of great importance for the behaviour of the tank. If the elastic analysis is used to determine the necessary reinforcement, severe cracking can occur in certain parts of the tank with resulting leakage. Figure 8 shows, as an example, the hoop force N_θ and the meridional moment M_φ in the linear and nonlinear case.

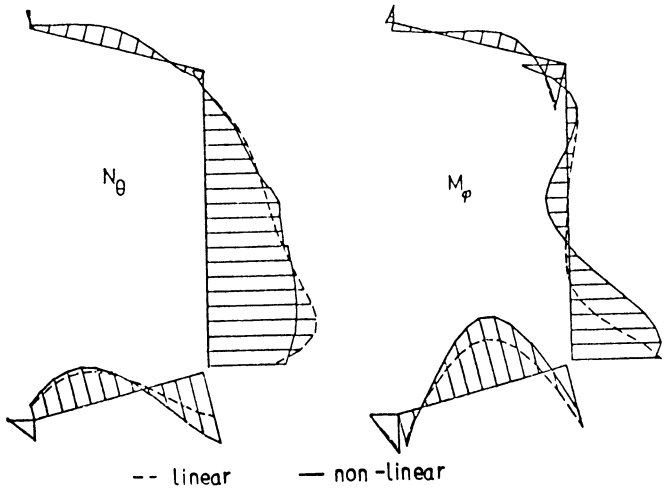


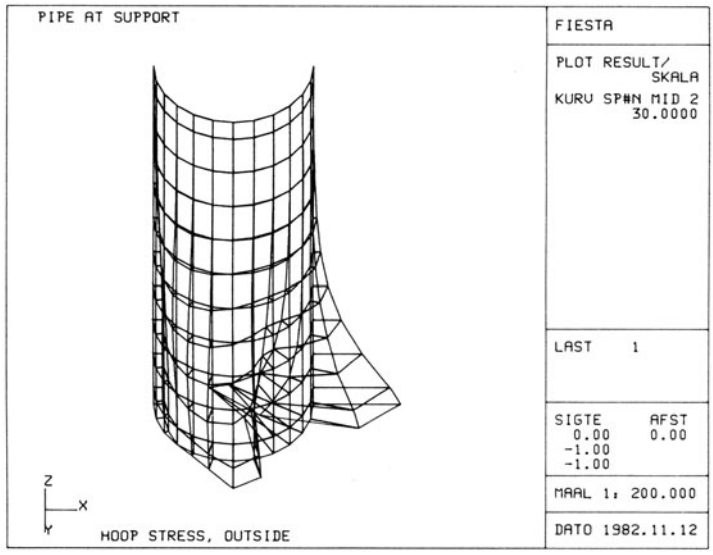
Figure 8. Linear and nonlinear response

EXAMPLES OF GRAPHICAL OUTPUT

This section shows further examples of the various possibilities for graphic presentation of results from FIESTA.

Stresses in circular pipe near support

A horizontal pipe is supported at regular intervals by two "feet" at an angle of 45° with the vertical plane of symmetry. The stress distribution is calculated by linear analysis.



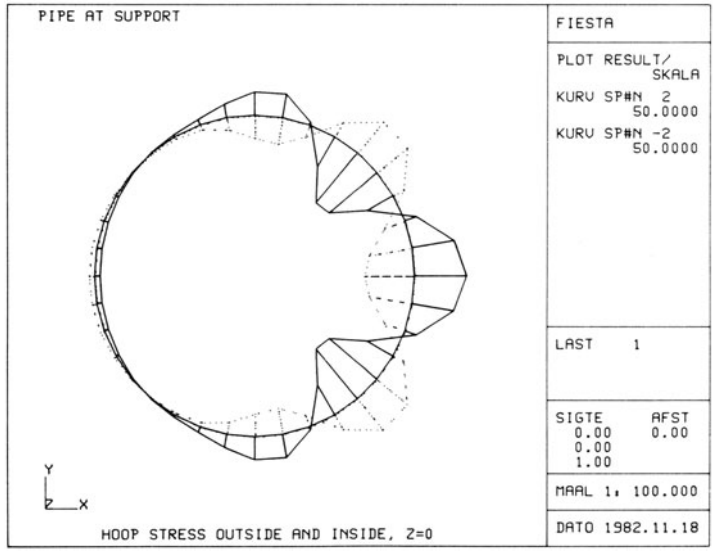
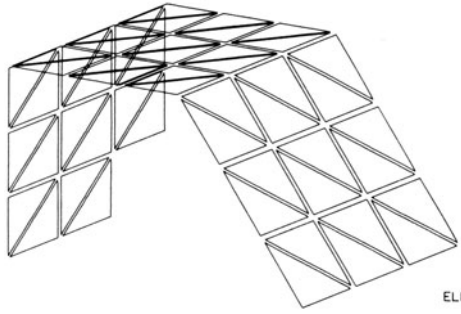


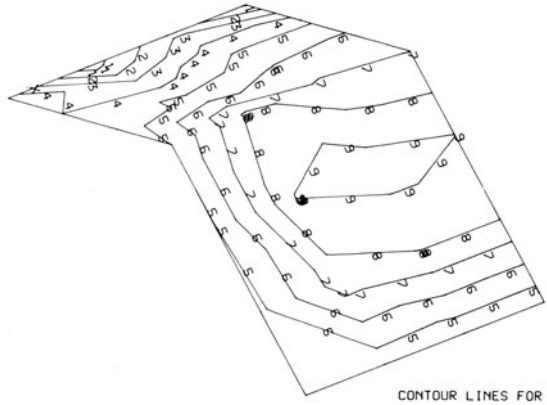
Figure 9. Sample plots, pipe near support

Tunnel with sloping wall

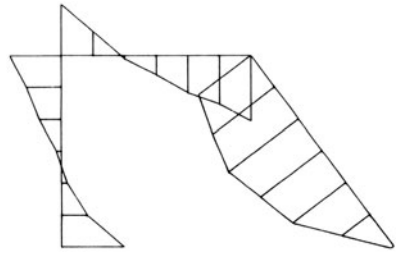
This example is meant for demonstration purposes only, hence a relatively coarse element net is chosen. The tunnel has an opening in the vertical wall at the vertical plane of symmetry. The sloping wall and the ceiling are subjected to a uniform vertical load, and the ceiling carries an additional line load.



ELEMENT MODEL, SHRINK PLOT



CONTOUR LINES FOR SIGMA_Y, INSIDE



BENDING MOMENTS M_X, Z=-3

Figure 10. Sample plots, tunnel with sloping wall

REFERENCES

Bergholdt, M. and Grodtkjær, E. (1981): FIESTA - Finite Elements in Structural Analysis. ADB-Udvalget (Danish Academy of Technical Science), Lyngby, Denmark.

Grodtkjær, E. and Bergholdt, M. (1982): Nonlinear Analysis of Shells of Revolution with Arbitrary Loading. Proc. Int. Conf. Finite Element Methods, Shanghai, China.

COMPUTATION OF PILE-HEAD IMPEDANCES BY THE TRANSFER MATRIX METHOD

A.K. Basu
Consultant Engineer
Brown & Root (U.K.) Ltd.,
London.

A. Gupta
Senior Scientific Officer
Indian Institute of Technology
Delhi.

INTRODUCTION

For the dynamic analysis of structures or machines supported on piles it is convenient to use the substructuring technique and to replace the soil-pile system by frequency-dependent impedances at the structure-pile interface. These impedances represent the dynamic stiffness and damping of the system under harmonic excitation. Analytical evaluation of these properties is possible only in certain idealized cases (Nogami and Novak, 1976; Novak and Nogami, 1977), and the more generally applicable finite element method could be quite expensive. An alternative method, which is approximate but has quite wide applicability, involves treating the soil surrounding the pile as a Winkler medium so that the soil reaction at a point on the pile depends only on the displacement of the pile at that point. With this technique it is possible to treat any end condition at the pile-tip, arbitrary variation of cross section with depth and layering of the soil medium.

In the discrete approach adopted by Penzien (1970) the soil is replaced by masses, springs and dashpots lumped at selected nodes on the pile. While the (static) spring characteristics are evaluated with reasonable accuracy using Mindlin's solution in the linear elastic range the determination of the equivalent soil masses and damping constants is not straightforward. This problem is, however, avoided in the approach developed by Novak (1974), in which the soil is replaced by frequency-dependent complex-valued impedances, the real part of which represents stiffness and the imaginary part damping. The damping arises due to both radiation of waves to the far field and hysteresis loss in soil. The soil inertia effects do not explicitly appear in the formulation being included in the impedance terms. These terms are evaluated on the basis of the plane strain assumption, i.e. by taking the medium as composed of independent thin horizontal layers subjected to

uniform harmonic motion, thus accounting for only horizontally propagating waves. Nondimensional stiffness and damping parameters for linearly viscoelastic soil with frequency-independent hysteretic type material damping have been presented by Novak et al (1978) for the various vibration modes taking the soil support to be continuous along the piles.

This approach, however, leads to zero static stiffness for the soil-pile system for both vertical and horizontal motion, since the soil layer stiffnesses vanish at zero frequency due to the assumption of plane strain condition. To remove this deficiency Novak (1980) proposed that a constant (static) value be taken for the soil stiffness upto the frequency at which this value equals the stiffness obtained from the plane strain assumption, and showed that this significantly improved the accuracy of the pile-head stiffnesses.

In an alternative approach for evaluating soil layer impedances proposed by Liou and Penzien (1980) the local value of the impedance, say, against horizontal movement is taken as the product of the local value of the corresponding static stiffness as found from Mindlin's equation and a frequency dependent shape function for horizontal motion, taken to be the same at all points along the pile. These shape functions for horizontal and vertical motions do not include the effect of material damping in soil. Thus, they may lead to erroneous results for low frequency excitations, when radiation damping effects are small and material damping assumes importance.

The procedures used by Novak (1978, 1980) and Liou and Penzien (1980) for evaluating soil-pile impedances with variable pile and/or soil properties involve generation of overall dynamic stiffness matrix of the system (which could be of large size if the number of variations is large), followed by static condensation. It is the purpose of the present paper to describe a technique based on the lumped parameter transfer matrix approach which does not require the generation of the impedance matrix of the pile segments (unlike in Novak's approach) or the assembly of the overall matrix for the system, but involves products of only small size matrices (4x4 for horizontal and 2x2 for vertical and torsional vibration) and is thus computationally very efficient.

THEORY

The proposed technique is based on the following assumptions:

(i) The soil medium is composed of discrete horizontal layers that extend to infinity and are homogeneous, isotropic and linearly viscoelastic with frequency-independent hysteretic type material damping. The soil properties are constant within each layer but may vary from layer to layer.

(ii) The pile is vertical and linearly elastic and its circular cross section is at least piece-wise uniform. The material damping in the pile is negligible.

(iii) The mass and the mass moment of inertia of the pile and the (complex-valued) vertical, horizontal, rocking and torsional spring stiffnesses provided by the soil along its length are lumped at suitably located nodes, so that the pile segments in between are massless and unsupported.

(iv) The soil layer below the pile tip is a viscoelastic half-space and its effect is represented by complex-valued frequency-dependent vertical, horizontal and rotational springs at the tip taking the tip as rigid for the purpose (Novak and Aboul-Ella, 1978).

(v) The vibration amplitudes are small and no separation takes place between the pile and the soil.

For evaluating the impedance functions the pile head is subjected in turn to harmonically varying vertical, horizontal, rotational and torsional displacements of unit amplitude and the system is analysed to find the steady state forces and moments developed at the pile head. Because of the assumptions of linearity these responses are also harmonic and have the same frequency as excitation.

For a given frequency of excitation the relevant soil dynamic stiffnesses (k) per unit length of the pile can be found from the following equations:

$$\text{Vertical} : k_w = G \left[S_{w1}(a_o, D) + iS_{w2}(a_o, D) \right] \quad (1)$$

$$\text{Horizontal} : k_u = G \left[S_{u1}(a_o, D, \upsilon) + iS_{u2}(a_o, D, \upsilon) \right] \quad (2)$$

$$\text{Rocking} : k_\psi = Gr^2 \left[S_{\psi 1}(a_o, D) + iS_{\psi 2}(a_o, D) \right] \quad (3)$$

$$\text{Torsional} : k_\theta = Gr^2 \left[S_{\theta 1}(a_o, D) + iS_{\theta 2}(a_o, D) \right] \quad (4)$$

where G is the shear modulus of the soil layer, $i = \sqrt{-1}$, S_{j1} and S_{j2} (j = w, u, ψ, θ) are respectively the non-dimensional stiffness and damping parameters in the direction of j, r the local outer radius of the pile, a_o the frequency parameter, D the damping ratio and υ the Poisson's ratio for the layer. a_o equals $\omega r/V_s$, where ω is the excitation frequency in radians/sec and V_s is the shear wave velocity in the soil layer being equal to $\sqrt{G/\rho_s}$, where ρ_s is the (wet) density of the soil. D equals half the damping constant $\tan\delta$, which gives the ratio of the imaginary and the real parts of the complex shear modulus of the soil.

The variations of the parameters S_{j1} and S_{j2} against a_o and $\tan\delta$ (= 2D) have been obtained by Novak et al (1978) and are shown in Figs. 1(a)-(d), the S_{u1} curves being valid only for υ = 0.5. As can be seen, S_{w1} and S_{u1} tend to zero as

a_0 tends to zero. In the present work this deficiency has been rectified by taking these parameters to vary linearly between a_0 = 0 and 0.3, from their static values at one end to the values given by Novak at the other. For a soil layer with υ = 0.5 these static values can be found for any pile slenderness ratio (H/r) from the equations given by Liou and Penzien (1980) and have been presented in Table 1. The stiffnesses have been expressed in terms of the Young's modulus of the soil so that they are approximately valid also for Poisson's ratios other than 0.5. The adjustment of the S_{w1} and S_{u1} curves shown in Figs. 1(a)and(b) are for H/r = 40 and 50 respectively.

Table 1. Variation of average static soil moduli (expressed in terms of Young's modulus) with slenderness ratio.

H/r	10	20	30	40	50	60	70	80	90	100
E_h/E	1.12	0.95	0.87	0.82	0.78	0.76	0.74	0.72	0.71	0.69
E_v/E	0.81	0.64	0.57	0.53	0.50	0.48	0.46	0.45	0.44	0.43

The spring stiffnesses provided to the pile tip by the soil medium below are found from the expressions given by Veletsos and Verbic (1973).

In order to use the lumped parameter idealization the soil-pile system is discretized into a number of constant property segments (not necessarily of equal length), and the pile mass and the soil stiffness are lumped at the nodes. These nodes are selected at sections where either the soil or the pile properties change, and also at a few locations in between in order to improve the accuracy of the solution, where necessary.

The discretized systems for vertical, horizontal and torsional vibrations are shown in Figs. 2(a)-(f). The lumped mass M_s and the lumped mass moments of inertia J_s about the horizontal axis and J_{vs} about the vertical axis at the node s are given by (for s = 1,2, ...n-1)

$$M_s = (m_s L_s + m_{s+1} L_{s+1}) / 2 \quad (5)$$

$$J_s = (m_s L_s^3 + m_{s+1} L_{s+1}^3) / 24 \quad (6)$$

$$J_{vs} = (m_s L_s r_s^2 + m_{s+1} L_{s+1} r_{s+1}^2) / 2 \quad (7)$$

in which m_s is the mass per unit length, L_s the length and

r_s the radius of gyration of the sth pile segment. The lumped soil stiffnesses at the node s (for s = 1,2,....n-1) are given by

$$K_{js} = (k_{js} L_s + k_{j(s+1)} L_{s+1}) / 2 \qquad (8)$$

where j is to be taken as w, u, ψ and θ for vertical, horizontal, rocking and torsional modes of vibration respectively and k_{js} is the soil dynamic stiffness per unit length of pile for the sth layer and is to be found from Equation 1, 2, 3 or 4. The lumped masses and stiffness at the end nodes (s = o and n) are found by taking the nonexistent segment lengths L_o and L_{n+1} as zero and adding at the tip node the appropriate stiffnesses of the bearing soil medium.

Under harmonic excitation the lumped parameter system may be considered as statically loaded if the inertia effects are included at each node by means of two translational springs of stiffness $(-\omega^2 M_s)$ and two rotational springs of stiffness $(-\omega^2 J_s)$ and $(-\omega^2 J_{vs})$. The elements of the pile-head impedance matrix for the excitation frequency ω then simply equal the pile-head forces and moments (P^a_{wn}, P^a_{un}, $P^a_{\psi n}$ and $P^a_{\theta n}$) developed due to a unit value of each of the displacments w_n, u_n, ψ_n and θ_n in turn. In the transfer matrix method used here the tip displacments (or forces) are given unit values in turn and the resulting forces and displacments above the node n are found by successive matrix multiplication. These coexisting forces and displacments at the pile head are then utilized to evaluate the elements of the impedance matrix. The main steps of the procedure are now presented for vertical, horizontal and torsional vibrations separately, since these are not coupled.

Vertical vibration (Figs. 2(a) and (b))

Considering the sth segment and noting that $w_s = w^a_s = w^b_s$, one gets the following relation for the state vectors at (s-1) and s:

$$\begin{Bmatrix} P_w \\ w \end{Bmatrix}^a_s = \begin{bmatrix} 1 + (L/EA)(K_w - \omega^2 M) & (K_w - \omega^2 M) \\ (L/EA) & 1 \end{bmatrix}_s \begin{Bmatrix} P_w \\ w \end{Bmatrix}^a_{s-1} \qquad (9)$$

in which (EA) is the axial rigidity of the sth segment and the (2x2) matrix is the transfer matrix involving properties of the sth segment and the sth node. Denoting this matrix by $[A_s]$ the pile-head force and displacment can be expressed in terms of those just above the node at the pile tip as

$$\begin{Bmatrix} P_w \\ w \end{Bmatrix}^a_n = [A_n][A_{n-1}] \ldots [A_2][A_1] \begin{Bmatrix} P_w \\ w \end{Bmatrix}^a_o \qquad (10)$$

If the pile tip is supported on a rigid stratum, w_o =

w_o^a = o and P_{wo}^a can be taken as unity. Otherwise, P_{wo}^b is taken as zero and w_o = w_o^b as unity; considering equilibrium at the tip node one then gets P_{wo}^a = $(K_{wo} - \omega^2 M_o)$ and w_o^a = 1. Equation (10) then gives P_{wn}^a and w_n, both of which are, in general, complex valued. The pile-head impedance for vertical vibration can now be found from

$$K_{ww} = P_{wn}^a / w_n = k_{ww} + i\,\omega c_{ww} \tag{11}$$

where k_{ww} is the real stiffness and c_{ww} is the coefficient of equivalent viscous damping.

Horizontal vibration (Figs. 2(c) and (d))

Considering the stiffness matrix of the segment s including instability effects and then the equilibrium of the node s, the following relation is obtained between the state vectors at (s-1) and s:

$$\begin{Bmatrix} P_u \\ P_\psi \\ u \\ \psi \end{Bmatrix}_s^a = \left[B_s\right] \begin{Bmatrix} P_u \\ P_\psi \\ u \\ \psi \end{Bmatrix}_{s-1}^a \tag{12}$$

The elements of the transfer matrix are given below (with suffix s omitted) in terms of the segment flexural rigidity (EI) and the stability functions S, C and m, which depend only on the ratio of the (static) axial load in the segment and its Euler load $\pi^2 (EI)_s/L^2{}_s$:

$$b_{11} = 1 + mCL^3 (K_u - \omega^2 M) / \left[EIS\,(1+C)Q\right]$$

$$b_{12} = mL^2 (K_u - \omega^2 M) / (EISQ)$$

$$b_{13} = -(K_u - \omega^2 M)$$

$$b_{14} = mL(1-C)(K_u - \omega^2 M) / Q$$

$$b_{21} = -\left[mL(1-C) + mL^2 (K_\psi - \omega^2 J)/EIS\right] / Q$$

$$b_{22} = \left[m(1+C)-2- 2L(K_\psi - \omega^2 J)/EIS\right] / Q \tag{13}$$

$$b_{23} = 0 = b_{43}$$

$$b_{24} = \left[2EIS(1-C^2)(m-1)/L + \left\{m(1+C)-2\right\} (K_\psi - \omega^2 J)\right] / Q$$

$$b_{31} = -mCL^3 / \left[EIS(1+C)\,Q\right]$$

$$b_{32} = -mL^2 / (EISQ) = b_{41}$$

$b_{33} = 1$

$b_{34} = -mL(1-C) / Q$

$b_{42} = -2L / (EISQ)$

$b_{44} = \left[m(1+C)-2\right] / Q$

with $Q = 2C - m(1+C)$. In segments where instability effects can be neglected one can take $S = 4$, $C = 1/2$, $m = 1$ and $Q = -1/2$.

The pile head forces and displacements can now be written in terms of those just above the node o as

$$\begin{Bmatrix} P_u \\ P_\psi \\ u \\ \psi \end{Bmatrix}_n^a = \begin{bmatrix} B_n \end{bmatrix}\begin{bmatrix} B_{n-1} \end{bmatrix} \cdots\cdots \begin{bmatrix} B_2 \end{bmatrix}\begin{bmatrix} B_1 \end{bmatrix}\begin{Bmatrix} P_u \\ P_\psi \\ u \\ \psi \end{Bmatrix}_o^a \qquad (14)$$

The nature of the horizontal vibration problem is such that two of the four forces or displacments at the bottom of the pile, i.e. below the node o, will be known from the specified support conditions. Each of the remaining two quantities will be given a unit value in turn while the other one is taken as zero, thus yielding two independent sets of forces/displacments at the pile head. Let these two sets be denoted by $\{P^a_{un1} \quad P^a_{\psi n1} \quad u_{n1} \quad \psi_{n1}\}$ and $\{P^a_{un2} \quad P^a_{\psi n2} \quad u_{n2} \quad \psi_{n2}\}$. Then, from the definition of the impedance matrix one has

$$\begin{bmatrix} K_{uu} & K_{u\psi} \\ K_{\psi u} & K_{\psi\psi} \end{bmatrix}\begin{bmatrix} u_{n1} & u_{n2} \\ \psi_{n1} & \psi_{n2} \end{bmatrix} = \begin{bmatrix} P^a_{un1} & P^a_{un2} \\ P^a_{\psi n1} & P^a_{\psi n2} \end{bmatrix} \qquad (15)$$

which yields the following relation for evaluating the impedance matrix:

$$\begin{bmatrix} K_{uu} & K_{u\psi} \\ K_{\psi u} & K_{\psi\psi} \end{bmatrix} = \begin{bmatrix} P^a_{un1} & P^a_{un2} \\ P^a_{\psi n1} & P^a_{\psi n2} \end{bmatrix}\begin{bmatrix} u_{n1} & u_{n2} \\ \psi_{n1} & \psi_{n2} \end{bmatrix}^{-1} \qquad (16)$$

It may be noted that all the elements in the above matrices are, in general, complex-valued. For convenience the elements of the impedance matrix can be written in the following form as was done in the case of vertical vibration

$$K_{pq} = k_{pq} + i\,\omega c_{pq} \quad (p,q = u, \psi), \qquad (17)$$

in which k_{pq} is the real stiffness and c_{pq} is the co-efficient of equivalent viscous damping. Also note that $K_{u\psi} = K_{\psi u}$.

Torsional vibration (Figs. 2(e) and (f))

Proceeding as in the previous cases the required relationship between the state vectors is

$$\begin{Bmatrix} P_\theta \\ \theta \end{Bmatrix}_s^a = \begin{bmatrix} 1+(K_\theta - \omega^2 J_v)L/(G_p I_p) & (K_\theta - \omega^2 J_v) \\ L/(G_p I_p) & 1 \end{bmatrix}_s \begin{Bmatrix} P_\theta \\ \theta \end{Bmatrix}_{s-1}^a \qquad (18)$$

where $(G_p I_p)_s$ is the torsional rigidity of the sth segment. The pile-head impedance for torsional vibration is obtained as in the case of vertical vibration and can be expressed as

$$K_{\theta\theta} = k_{\theta\theta} + i\omega c_{\theta\theta} \qquad (19)$$

NUMERICAL RESULTS

To verify the proposed technique for evaluating impedance functions the following cases involving a solid pile of uniform cross section and embedded in a uniform soil stratum were analysed and the results compared with those available in the literature (Novak, 1974: Novak and Aboul - Ella, 1978; Novak et al, 1978).

(i) Vertical vibration: $H/r = 40$, $V_s/V_c = 0.03$, $\rho_s/\rho_p = 0.7$, $\tan\delta = 0$, and $\upsilon = 0.4$, where $V_c = \sqrt{E_p/\rho_p}$, E_p being the Young's modulus and ρ_p the mass density of the pile material. The pile is assumed end bearing on a rigid stratum.

(ii) Horizontal vibration: $H/r = 50$, $V_s/V_c = 0.03$, $\rho_s/\rho_p = 0.6$, $\tan\delta = 0.1$ and $\upsilon = 0.25$. The pile is considered to be of the floating type.

(iii) Torsional vibration: $H/r = 70$, $V_s/V_c = 0.02$, $\rho_s/\rho_p = 0.7$, $\tan\delta = 0$ and 0.1, $\upsilon = 0.25$. The pile tip is assumed fixed.

For the vertical vibration case the static stiffness parameter for the soil, S_{w1} ($a_0 = 0$), was found from Table 1 to be 1.48. Solutions were obtained using 10, 20 and 30 segments of equal length and the results for 20 segments were found to be within 0.5% of those for 30 segments. The real and the imaginary parts of the impedance function are shown in Fig. 3 and are found to be in excellent agreement with the Novak solution. These results also agree with the analytical solution obtained by treating the pile as a bar on elastic foundation having a complex-valued stiffness per unit length of $\bar{k}_w = (GS_{w1} - \omega^2 m) + i(GS_{w2} + \omega \bar{c}_w)$, where $\bar{c}_w$ is the viscous damping constant of the pile per unit length. ($\bar{c}_w$ has been taken as zero in the present work). This gives the pile-head impedance, for an end bearing pile of length H:

$$K_{ww} = (\mu EA) \coth(\mu H). \qquad (20)$$

in which $\mu = \sqrt{\bar{k}_w / (EA)}$. The impedance for an infinitely long pile is then $\sqrt{\bar{k}_w\ EA}$. The real and imaginary parts of this impedance function are also shown in Fig. 3.

For the pile under horizontal vibration the soil stiffness parameter S_{u1} at a_o=0 was found to be 1.96 from Table 1. The effect of axial loading on pile stiffness was neglected. Results were again obtained for 10, 20 and 30 equal segments and 20 divisions were found to give sufficient accuracy. Figure 4 shows that the present solution is in very good agreement with the Novak solution, and also with the analytical solution for an infinitely long beam on elastic foundation. The foundation stiffness $\bar{k}_u$ is in this case $(GS_{u1} - \omega^2 m) + i(GS_{u2} + \omega \bar{c}_u)$, where $\bar{c}_u$ is the viscous damping constant per unit length of the pile for horizontal vibration and is assumed zero in the present work. The three relevant impedance functions are

$$K_{uu} = 4\beta^3 EI \tag{21}$$

$$K_{u\psi} = K_{\psi u} = 2\beta^2 EI \tag{22}$$

$$K_{\psi\psi} = 2\beta EI \tag{23}$$

in which $\beta^4 = \bar{k}_u/(4EI)$. As is evident from the expression for $\bar{k}_u$, the rotatory inertia of the pile and the rocking stiffness and damping (S_ψ) provided by the soil layers have been ignored in deriving the analytical expressions for the impedance functions.

For the pile under torsional vibration the results are shown in Fig. 5 and are found to be in excellent agreement with the Novak solution. These results where obtained using 40 equal segments, the values for 30 segments being within 0.5% of these values.

TREATMENT OF SOIL NONLINEARITY

For a practical, design office analysis of dynamic interaction between superstructure and soil-pile system in the presence of relatively large pile movements, use can again be made of pile-head impedances. Impedances in such cases have to correspond to appropriate levels of the harmonically varying force- or displacement- excitation at the pile head. Using the present modelling the soil layers are again taken as linearly viscoelastic, but their G and D values now vary with the strain or displacement level in them. Thus, although the curves for layer impedance parameters S_{u1} and S_{u2} etc. (Figs. 1(a)-(d)) remain valid, their actual values for a given frequency now depend on the amplitude of excitation. (Note that a change in G affects a_o through V_s.).

The relationships between the vibration amplitude on the one

hand and the shear modulus and damping of soil layers on the other are thus the basic data required for the proposed analysis. Since these data have not yet been developed, the following equations in nondimensional form are suggested as approximations, which permit the use of the nonlinear soil support curves available in the literature:

$$G/G_{max} = E_s/E_{so} \tag{24}$$

$$D/D_{max} = 1-G/G_{max} \tag{25}$$

where G_{max} is the soil modulus corresponding to small strains, D_{max} the limiting value of the damping ratio occuring at large strains, E_s the static layer stiffness at specified displacment and E_{so} its small-displacement value. Equation (24) can be considered to be at least qualitatively valid, and Equation (25) is based on the results of numerous tests (Hardin and Drnevich, 1972). In the above, E_s is to be interpreted as the secant stiffness.

As an example consider the problem of horizontal vibration of a steel pipe pile analysed by Angelides and Roesset (1980) using the finite element method (FEM). The pile is embedded in a clayey soil, has a floating tip and is subjected only to a horizontal load at its top. It is of uniform cross section with 1.22m outside diameter and 63.5mm wall thickness and has a length of 54.9m. The soil is also of uniform properties with G_{max} = 15,700 KN/m^2, D_{max} = 0.285, ρ_s = 1800 Kg/m^3, υ = 0.5 and $E_{so} = 2.12G_{max}$ (from Table 1). The additional soil properties needed for nonlinear analysis are: C_u (undrained shear strength) = 96.1 KN/m^2; γ_b (buoyant unit weight) = 8.02 KN/m^3; y_c (deflection corresponding to soil resistance of half the peak value, p_u) = 0.0305m; Z_r (depth below which failure occurs by plastic flow in horizontal planes) = 12.16m.

E_s vs y relations for layers at different depth are derived from the p-y curves for cyclic loading given by Matlock (1970). Since E_s so derived tends to infinity as y tends to zero, the E_s - y curve for small values of y has been replaced by its tangent that passes through E_{so} at y = o. (This, incidentally, gives a value of $E_s = 0.6E_{so}$ at the point of tangency, where $y = \sqrt{(5p_u/6E_{so})^3/y_c}$.)

For a specified frequency the pile-head impedances with zero moment condition and the associated deflection profile are first evaluated for small amplitude vibrations by taking for each layer G = G_{max} and D = O. Note that S_{u1} (a_o=o) = 2.12. Then, to find the solution for a specified displacement amplitude at the pile head, the displacements at the other nodes are assumed on the basis of this deflected shape and, hence, the values of E_s, G and D are determined (Equations (24) and (25)). This defines the layer impedances, from which

the pile head impedances and the associated deflected shape are calculated. If the computed shape is not close enough to the one assumed, a fresh calculation is performed on the basis of the new profile, and the process continued till convergence. The converged deflection profile is then used for the next increment of the pile head displacment. Finally, the amplitude of the pile-head horizontal force associated with a displacment is found by multiplying the displacment with the absolute value of the impedance.

Some of the results of this computation are presented in Figs. 6 and 7. Figure 6 shows how the stiffness and damping of the soil layers at the ground level and at a depth Z_r below the ground vary with the frequency and amplitude of vibration. The deterioration in both stiffness and damping with amplitude is clearly seen, as also its severity for the layer at the ground level. For comparison the real and imaginary parts of the equivalent layer impedances for small-amplitude vibration implicit in the FEM solution are also shown. These were derived from the linear pile-head impedances and noting that, according to the Winkler model, they are related to the layer impedances by the equation

$$K_{uu} = 2\,\beta^3\,EI \qquad (26)$$

where β is a function of layer impedance as defined earlier. It appears that when compared with the present approach the FEM model exhibits much higher stiffnesses but a lower damping. To have a meaningful validation of the present technique against the FEM solution, impedances were computed using values for E_{so} and E_s which were 2.3 times those found from the p-y curves and S_{u2} values equal to 0.6 times those given by Novak. These results are compared with the FEM solution in Fig. 7. Agreement between the two sets is found to be good.

CONCLUSION

The technique presented in the paper for evaluating impedance functions of linear soil-pile systems has been found to be computationally very efficient. This coupled with the capability to tackle soil layering, nonuniform pile cross section, instability effects in piles (where necessary) and a variety of support conditions makes the technique a powerful tool for routine use in the design office.

A method has also been outlined for evaluating approximate pile impedances taking into account soil nonlinearity. This makes use of certain hypotheses regarding the reduction in G and increase in D with the level of soil deformation and the cyclic p-y or t-z curves. The method produces as limiting cases the nonlinear pile-head static stiffness curve and the linear impedance functions.

ACKNOWLEDGEMENTS

The authors would like to thank the Director of Indian Institute of Technology, Delhi (where most of the work reported here was originally carried out) and the Management of Brown & Root (UK) Ltd, London for extending to them necessary facilities for the preparation of this paper.

REFERENCES

Angelides, D.C. and Roesset, J.M. (1980) Nonlinear Dynamic Stiffness of Piles. Research Report R 80-13, Massachusetts Institute of Technology, Deptt. of Civil Engg., April: 1-36.

Hardin, B.O. and Drnevich, V.P. (1972) Shear Modulus and Damping in Soils: Design Equations and Curves. Jour. of the Soil Mechanics and Foundations Div., ASCE, 98, SM7, July: 667-692.

Liou, D.D. and Penzien, J. (1980) Mathematical Modelling of Piled Foundations. Numerical Methods in Offshore Piling, Institution of Civil Engineers London: 69-74.

Matlock, H. (1970) Correlations for Design of Laterally Loaded Piles in Soft Clay. Proc. of the 2nd Offshore Technology Conference, Houston, Texas: 577-594.

Nogami, T. and Novak, M. (1976) Soil-pile Interaction in Vertical Vibration. Int. Jour. Earthq. Engg. and Struct. Dynamics, 4, 3: 277-293.

Novak, M. (1974) Dynamic Stiffness and Damping of Piles. Canadian Geotech. Jour., 11, 4: 574-598.

Novak, M. and Nogami, T. (1977) Soil-pile Interaction in Horizontal Vibration. Int. Jour. Earthq. Engg. and Struct. Dynamics, 5, 3: 263-281.

Novak, M. and Aboul-Ella, F. (1978) Impedance Functions of Piles in Layered Media. Jour. of the Engineering Mechanics Div. ASCE, 104, EM3, June: 643-661.

Novak, M. et al. (1978) Dynamic Soil Reactions for Plane Strain Case. Jour. of the Engineering Mechanics Div., ASCE, 104, EM4, August: 953-959.

Novak, M. (1980) Soil-pile Interaction under Dynamic Loads. Numerical Methods in Offshore Piling, Institution of Civil Engineers London: 59-68.

Penzien, J. (1970) Soil-pile Foundation Interaction. Earthquake Engineering; (ed. Wiegel R.L.), Prentice-Hall; Englewood Cliffs, NJ, Chapter 14, 349-381.

Veletsos, A.S. and Verbic, B. (1973) Vibration of Viscoelastic Foundations. Int. Jour. Earthq. Engg. and Struct. Dynamics, 2, 1: 87-102.

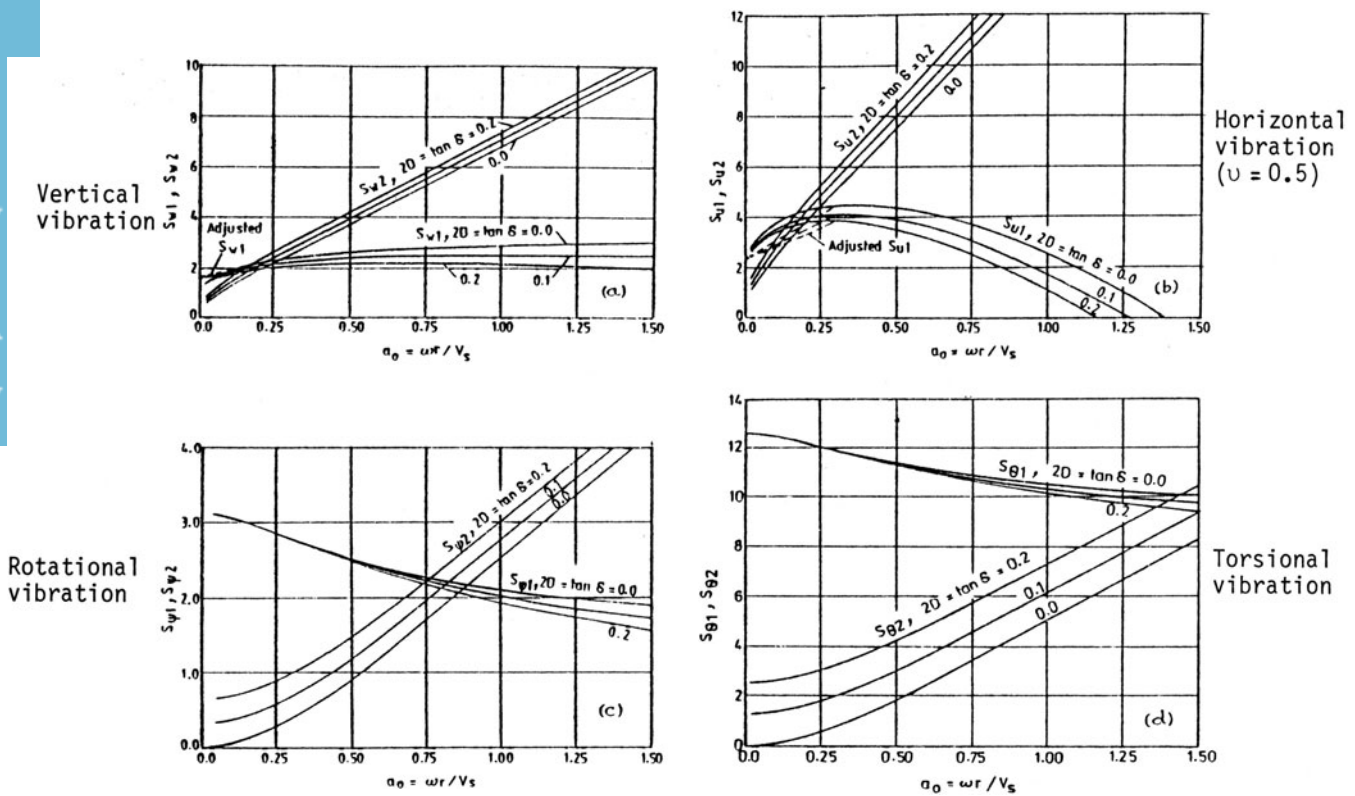


Figure 1. Stiffness and damping parameters for viscoelastic layers under plane strain.

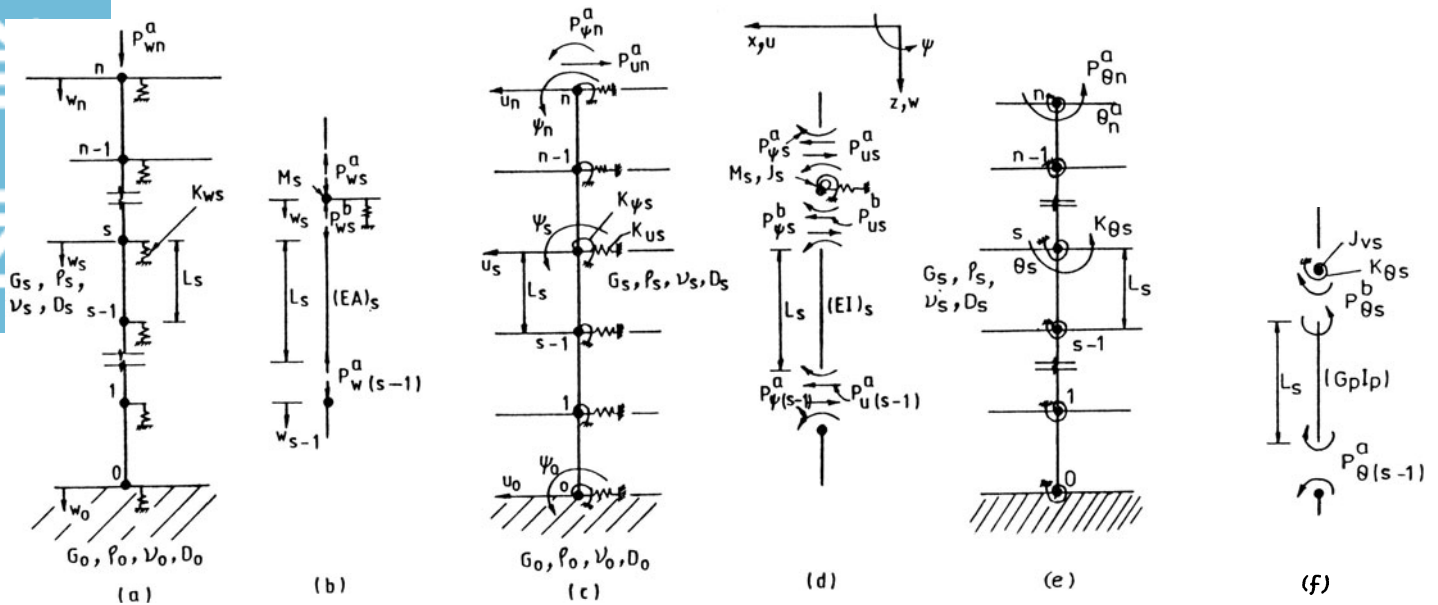


Figure 2. Model of pile subjected to vertical, horizontal or torsional vibration.

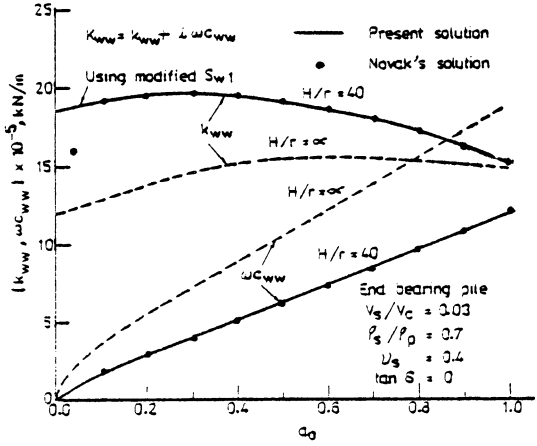


Figure 3. Comparison of present solution with Novak's solution for vertical impedance function.

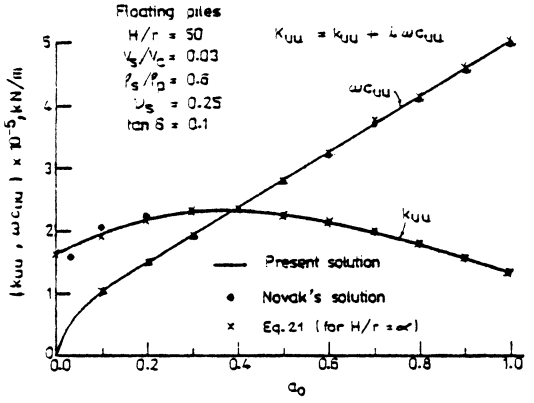


Figure 4. Comparison of present solution with Novak's solution for horizontal impedance function.

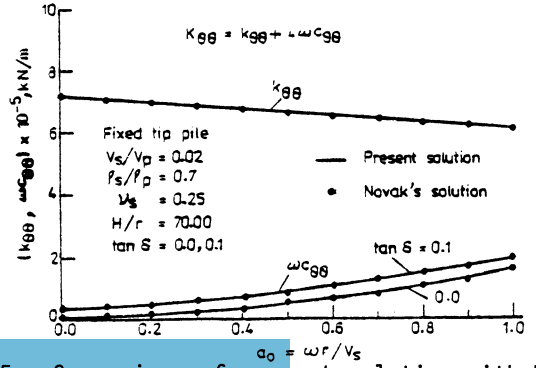


Figure 5. Comparison of present solution with Novak's solution for torsional impedance function ($V_p = \sqrt{G_p/\rho_p}$)

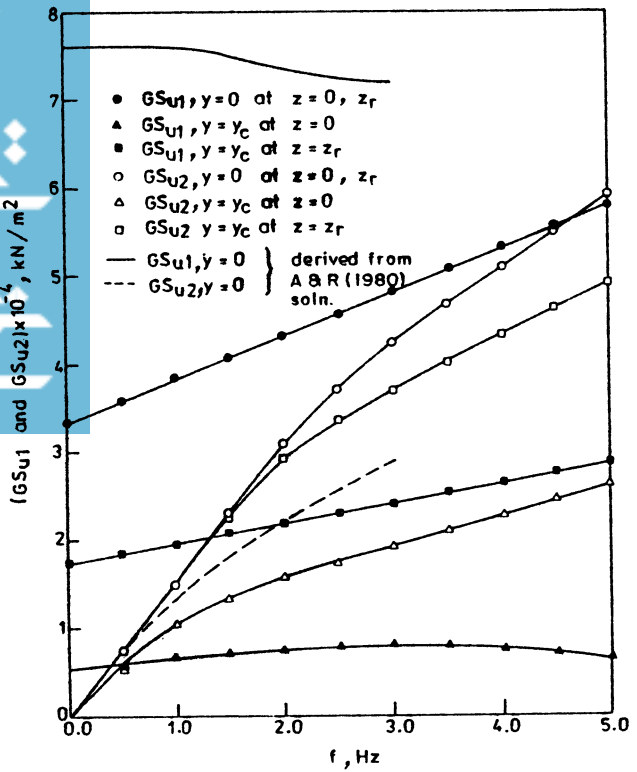


Figure 6. Variation of GS_{u1} and GS_{u2} with frequency and displacement amplitude.

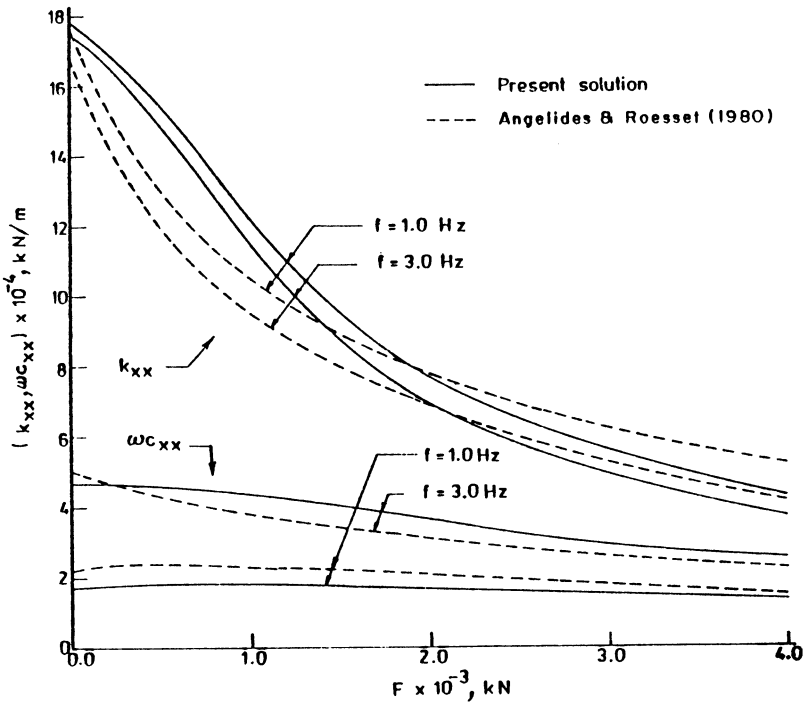


Figure 7. Validation of present solution for pile-head lateral stiffness and damping.

BILLS OF QUANTITIES AT A STROKE

J.J. Gibson*, J.C. McIlwrath*, M.I. Reid†, W.R.H. Barton†

*Queen's University of Belfast
†Kirk, McClure and Morton - Consulting Engineers

INTRODUCTION

These introductory paragraphs outline some aspects of the function and form of Bills of Quantities in order to provide a working perspective to the main body of this paper. They have been deliberately confined to a number of pragmatic considerations directly related to the preparation and assembly of Bills of Quantities and do not touch upon philosophical, legal or contractual aspects of contract documentation.

Bills of Quantities

Bills of Quantities are used throughout the construction industry as a basis for financial models of construction projects and as a mechanism for their financial control. These ends are normally achieved through the systematic processes of:

(i) Identifying, describing and listing the permanent (and temporary) Works items of a project.
(ii) Obtaining tenders for the project on the basis of the list of items and their Quantities.
(iii) Conducting a comparative analysis of tenders received.
(iv) Constructing a financial model of the project on the basis of tenders received to establish its probably cost and expenditure profile.
(v) Taking the decision to proceed (or not) with the project or to modify it in the light of the financial data now available.
(vi) Establishing the value of work executed and controlling payments as work on the project proceeds.

Most Bills of Quantities are highly structured documents which make extensive use of specialised terminology. Their structure, content and interpretation is based upon accepted conventions contained in a set of rules and definitions referred to as the "Method of Measurement".

Method of Measurement

Both the Bill of Quantities and the Method of Measurement on which it is based are normally defined in a document called the "Conditions of Contract" which is a statement of the contractual agreements and arrangements between an Employer and a Contractor.

In the course of time, within the various sections of the construction industry, a number of different Methods of Measurement have been evolved to meet the needs and circumstances of the different types of work undertaken and the variety of contractual arrangements utilised. This evolution is a continuing process and the Professional Bodies, associated with the industry, review and revise their Methods of Measurement when such action seems warranted.

The "Civil Engineering Standard Method of Measuring"

In 1976, the Institution of Civil Engineers and the Federation of Civil Engineering Contractors - in association with the Association of Consulting Engineers, published a document entitled the "Civil Engineering Standard Method of Measurement" (CESMM) for use in Works of Civil Engineering Construction[3]. This document superseded the former "Standard Method of Measurement for Civil Engineering Quantities" which had been in use (with modifications) since 1955[4].

The amendments to Clause 57 in the 1979 revision of the I.C.E. Conditions of Contract[5] gave formal recognition to the general use of the CESMM in Civil Engineering Contracts.

The first formal reference to computer-aided techniques in Civil Engineering contract documentation appears in the foreword to the CESMM where, in referring to the system of work classification which had been adopted, it was suggested that this should "make the use of computers easier". Not a world-shattering statement perhaps, but it gave a strong indication of the direction in which the Institution of Civil Engineers wished its members to proceed. This point was not further developed and no further reference to computers is contained in the CESMM.

In 1977, subsequent to the publication of the CESMM, the Institution of Civil Engineers published a book entitled "Measurement in Contract Control"[1] which contained a closely-reasoned exposition of the philosophy and principles on which the CESMM was based and some examples of its use were set out in the text. This book, despite the disclaimer contained therein, has become widely accepted as the definitive document in respect of the use and interpretation of the CESMM.

In Section 3 of this book, the author, Dr. N.M.L. Barnes, outlined his vision of a future in which extensive use will be made of computers in the preparation and pricing of Bills of Quantities. His comments in this respect were, of necessity, generalised, but their intention is clear - the CESMM has been

specifically designed to facilitate the use of computers in the preparation and pricing of Bills of Quantities.

The new features introduced by the CESMM which are of most relevance in this respect are:-

(i) the standardisation of layout of Bills of Quantities;
(ii) the structured system of work classification;
(iii) the set of rules for the wording of item descriptions;
(iv) the alpha-numeric code system for item references.

Manual Processes of BoQ Preparation

To determine how computers might best be used in the preparation and production of Bills of Quantities (BoQ) in accordance with the CESMM a first step was to examine the existing manual (non computer-aided) processes by which Bills of Quantities are produced and to identify areas where time and effort could be reduced by the application of computer-aided techniques.

To avoid embarking upon a tedious exposition of the processes by which a typical Bill of Quantities is manually prepared and produced it will be necessary to accept that these processes can be time consuming.

The following brief outline of a typical sequence of operations should, however, provide a sufficient description of the nature of the processes.

(i) an engineer prepares a manuscript draft BoQ based upon the contract drawings and specification;
(ii) a typist produces a typed draft of the manuscript;
(iii) an engineer checks the typed draft against the manuscript draft; makes corrections and amendments to the text;
(iv) a typist incorporates corrections and amendments to typed draft (possibly produces a re-typed draft);
(v) an engineer checks the re-typed draft, probably incorporates late additions and omissions and probably requests a more aesthetically pleasing re-arrangement of the typed text;
(vi) operations (iv)-(v) may be repeated several times until all the corrections, additions, omissions and re-arrangements have been effected;
(vii) the typescript is paginated;
(viii) the final typescript is photocopied, assembled and collated.

It should be noted that this outline sequence of operations largely avoids reference to the qualitative processes by which an engineer identifies and describes the nature of the work items and the quantitative processes whereby their magnitude is determined. Furthermore, prior to the advent of the rules for the wording of item descriptions in the CESMM an abundant spring of quasi-scientific terminology was available to the compiler of

a BoQ. The admonition to "drink deep or touch not" was well founded, for a little learning in this respect was indeed a dangerous thing as semantics in relation to Civil Engineering Contracts have put silken suits on the backs of lawyers.

From the foregoing and accepting the premises that:

(a) circumstances beyond the control of the engineer often enforce changes in the contents of a BoQ; and
(b) that textual amendments are most conveniently identified when a Bill of Quantities is in typescript form;

- the areas in which computers may be employed with advantage are in the compilation and processing of text.

Computer Program Specification

It was considered that a computer program to aid the compilation and processing of BoQ text should incorporate the following facilities:

(i) a mechanism for the visual display of a "ruled BoQ page" on which text can be assembled either by direct input from a typewriter keyboard or by recall from a storage medium;
(ii) the ability to edit displayed text by means of:
  (a) insertion, deletion and replacement of characters and lines with the automatic re-adjustment of succeeding text all within the columns of the BoQ page format
  (b) tab, left-justification and word "wrap-around" features for BoQ page format
  (c) "roll-up" and "roll-down" features;
(iii) the ability to store and recall text in conceptual units of A4 size pages in "job files" which can be accessed by page numbers and job numbers;
(iv) the automatic sequential numbering of pages in a job file as text is added or removed;
(v) the ability to construct and edit a library of item descriptions and headings in standard terminology referenced by the CESMM code numbering system for use in assembling BoQ text;
(vi) paper copy output of text by dot matrix and/or daisy wheel printer.

It was considered that the above schedule should provide the minimum basis for a first-level operational Bills of Quantities assembly program which would later be further developed to incorporate additional degrees of sophistication in facilities and scope.

The following operational requirements were considered essential:

(i) program lines and data should be protected against accidental corruption through user error;
(ii) the program should be fully interactive and "friendly";
(iii) all displayed instructions and questions should be worded in simple standard English;
(iv) the user should know the present mode of operation at all times by an inverse-video display on some fixed area of the V.D.U.;
(v) other available modes should be on display at all times;
(vi) a "Help" mode should be available at all times;
(vii) changes in mode should be by a single key-stroke on a dedicated (soft) key.

It was felt that further requirements would become apparent in the light of operational experience and that this would be a normal feature of program development.

DESIGN PHILOSOPHY AND IMPLIMENTATION

In addition to the specification set out above, two further constraints were imposed on the project:

(i) the choice of computer was dictated by the fact that Kirk, McClure and Morton's computer was a Hewlett-Packard 9845 (T), known as the "System 45".
(ii) H.P. Basic is the standard programming language of the System 45.

In order to allow the program suite (shown diagrammatically in Fig. 1) to be tailored to a variety of hardware, it has been written in 'modules', which can be removed or altered. Some research was carried out in the early stages of development regarding these modules. It was discovered that a subroutine can be executed much more quickly than an equivalent subprogram. Consequently, it was decided that subroutines should be the constituents of the modules because the program must be capable of keeping pace with the typist.

Hardware Environment

As stated the computer on which the program must operate is a Hewlett-Packard System 45 with printing and storage peripherals.

Paper copy Two methods of producing a paper copy are available:

(i) an integral thermal printer which produces acceptable fast, rough copies. However, thermal printer paper is relatively expensive and the print is subject to deterioration.

(ii) a daisy wheel printer which produces high quality copies. It is envisaged that most users will select this option.

Storage media Two possible storage media for use with the H.P. System 45 are:

(i) Cassette tape With a single tape drive, both the program suite and the stored data must be accommodated on a cassette tape of 217K capacity. Since the program suite occupies approximately 100K in total, 117K is available for data.

(ii) Floppy discs An 8" dual disc drive provides much greater storage with the distinct advantage of faster access to both program and stored data. Furthermore, disc storage provides the opportunity of using Binary Data Files, which can be accessed at direct memory access (D.M.A.) rates. The Hewlett-Packard Dual Disc drive provides 2.3 M bytes of removable mass storage, i.e. enough capacity to hold the entire suite and approx. 275 BoQ pages.

The initially greater cost of the latter is considered to be more than offset by:

(a) its speed of operation
(b) its much greater storage capcity.

Hence, the 9845 (T) (187K, twin tape drives and thermal printer) complete with dual disc drive and daisy wheel printer is considered to provide an excellent hardware configuaration.

Software Environment

The Cursor Control of the cursor was, logically, assigned to the four vertical and horizontal arrow keys. Initially, the H.P. System 45 absolute cursor was utilised. This was eventually abandoned - as it was incapable of keeping pace with the typist - in favour of a binary routine. The cursor, upon reaching the boundary of a tab block, will be directed back to the opposite extremity of the same line i.e. a "wrap-around" facility.

Page composition and storage It was proposed that each character on a line in a page should be an element of a string (line). Several of these strings (lines) should form a string array (page). This proposal has been implemented and has proved very satisfactory.

The line capacity of a page exceeds the 20 lines which the VDU can display and it is therefore necessary to roll up/down in order to read a full page. This was achieved by the use of the same binary routine after insurmountable problems were encountered when using the screen memory. In operation, the cursor moves horizontally along the centre of the VDU while the page rolls "behind" it.

In order to minimize delays while the hardware accesses a page, three pages are held simultaneously in the computer's memory,

with the screen forming a window over the centre page. This means that the end of one page and the start of the next can be simultaneously displayed without delay.

Pages are automatically numbered sequentially. By default, page numbering is in Arabic numerals, however, the user may elect to operate in Roman numerals.

Since pages must be rapidly accessible, they are stored as Binary Data Files, which have the important advantage of operating at DMA rates. Hence each page is stored as an individual file.

Page Formats Initially, the program was developed in accordance with the BoQ page format. However, with modifications, this program would function as a general purpose word processor in order to generate other contract documents. Therefore, a plain page mode has been incorporated.

A method of storing these two modes was sought and this resulted in two alternatives:

(a) store the system as one large program
(b) store each of the two program modes as a separate program.

After some thought (a) was rejected because changing the mode within the large program would necessitate a large number of changes to parameters associated with the editing functions.

Therefore, two modal options now exist - BoQ and plain page format. The user has the option of transferring directly from one format to the other at any time. In either mode, the user may manipulate text by:

1 entering text directly from the keyboard:
   (a) using the CESMM Library
   (b) using the Glossary
   (c) entering characters individually.
2 altering existing text using the editing facilities.

Character transfer An important facility available during typing is character transfer. In the event of the cursor reaching the right hand tab setting with only part of a word entered, the system will automatically transfer the existing characters to the left hand side of the following line. Simultaneously, the user continues to enter the remaining characters of the word which will be concatenated to the existing characters.

Libraries There are two libraries provided by the program suite:

(a) a Library of item descriptions (CESMM Library)
(b) a Glossary of commonly used phrases.

Text from the CESMM Library is accessed and displayed by typing the CESMM item code, pressing [TAB] followed by the Special Function Key K1. The system automatically searches for the text corresponding to that code number and displays the item description.

Text from the glossary library is similarly accessed and displayed using the glossary codes (similar in structure to the CESMM codes). The required phrase will then be displayed, commencing at the current cursor position.

Display Area

The VDU on the System 45 contains 23 accessible lines, the bottom three of which have been assigned to a "Display Area". This area has been programmed to provide the user with information regarding the present function of the Special Function Keys, the current mode, page number, row number, column number and tab settings.

The Editor

The keys on the Hawlett Packard System 45 are divided into the following categories:

1 Alphanumeric keys
2 Display keys
3 Edit/System Command keys
4 Numeric keys
5 Program Control keys
6 Typewriter keys
7 Special Function keys.

The functions of these keys are described below:

1 Alphanumeric keys Both the layout and function of these keys are similar to those on a standard typewriter. However the following are non-standard keys in this block.

[TAB] positions the cursor in the leftmost corner of the next tab block.

[STORE] performs the function of RETURN on a typewriter. (Positions the cursor on the left-hand side of the next line of the current tab block.)

[REPEAT] and an alphanumeric key pressed simultaneously, repeatedly enters the character on the screen.

[CONTROL] and one of the designated alphanumeric keys depressed simultaneously will display a special drawing character.

2 Display keys These keys are used to move the cursor left or right, move the "page" up or down, clear the "page" contents completely, and clear all characters from the cursor position to the end of the line.

3 Edit/System Command keys The eight keys within this block have been programmed to operate:

[DEL LN] deletes the line upon which the cursor is stationed and subsequently moves succeeding text up one line. The cursor is then stationed on the left-hand side of the next line.

[INS LN] inserts a line immediately above the current cursor position. The cursor is then stationed on the left-hand side of the newly created line.

[INS CHR] inserts characters to the left of the cursor and causes the succeeding characters in the current paragraph to be advanced one space. The cursor then moves one space to the right, awaiting further character insertions. Control is transferred out of "Insert Character" mode when [INS CHR] is pressed again.

[DEL CHR] deletes the character at the current cursor position. The cursor remains in the same position and the remainder of the text on the line moves one character space to the left as each character is deleted.

[STEP] "steps" forward through displayed text, 10 lines at a time.

[PRT ALL] if this key is latched, a paper copy of each page "stepped" through will be printed.

4 Numeric keys The keys in this category operate in a similar manner to the numeric keys contained within the Alphanumeric block, with the exception of [RES], [CLEAR LINE] and [EXECUTE].

[RES] calculates and inserts the amount in the "Amount" column of a BoQ, provided that the "Rate" and "Quantity" data are already entered.

[CLEAR LINE] clears an entire line within the current text block.

[EXECUTE] is undefined as yet.

5 Program Control keys

[STOP] stores all pages currently in machine memory and stops execution.

[PAUSE] is undefined as yet.

[CONT] has the usual functions (continue).

6 Typewriter keys

[TAB SET] this key is only operational if the user is in plain page mode. Upon depressing this key, a tab is 'set' at the current cursor position. Only two tab may be set: specifying the left and right-hand extremities of a text block. If the tabs are not 'set' then they become columns 2 and 80 by default.

[TAB CLEAR] clears both tab settings, reverting to default tab settings.

[TYPEWRITER] has been disabled - program defines "Typewriter" mode.

7 Special Function keys Thirty-two keys are available, of which only KO-K9 have been defined at present. Their functions are:

KO - accesses Glossary
K1 - accesses CESMM Library
K2 - creates an empty plain page in the storage medium
K3 - creates an empty BoQ page in the storage medium
K4 - entirely deletes a page from the storage medium
K5 - retrieves a page from the storage medium
K6 - saves a specified page in the storage medium
K7 - prints a paper copy of the page currently displayed
K8 - prints a paper copy of the whole BoQ
K9 - prints a paper copy of any requested page.

USER REACTIONS

User reactions fall into two categories:

(i) those of the typist;
(ii) those of the engineer.

The typist Initally there was an undefined reluctance to sit at the computer keyboard and try the system. However, once sufficient assurance had been given that the typist could not (with anything approaching normal usage) wreck or destroy expensive hardware and software this initial reluctance was largely overcome.

Negative comments concerning the "feel" of the keys and the

(apparent) complexity of the array of keys soon ceased when it was discovered that the system behaved like a conventional typewriter.

Familiarisation with the special function keys required a little effort in the early stages. Again, as soon as it was appreciated just how simple they were to operate their use became accepted as no more than a conventional typing operation.

The Engineer In the case of the engineer, there seemed to be very little initial negative reaction.

Engineers have highlighted advantages:

1 The correction, addition and removal of text from documents in the process of preparation is fast, clean and easy. Printed output is not marred by erasures.

2 Engineers can do their own editing. This is most useful when typists are unavailable and saves much frustration.

3 Bills of Quantities can be assembled by directly plagarising an existing Bill of Quantities of similar content to that required and amending the quantities and descriptions as appropriate.

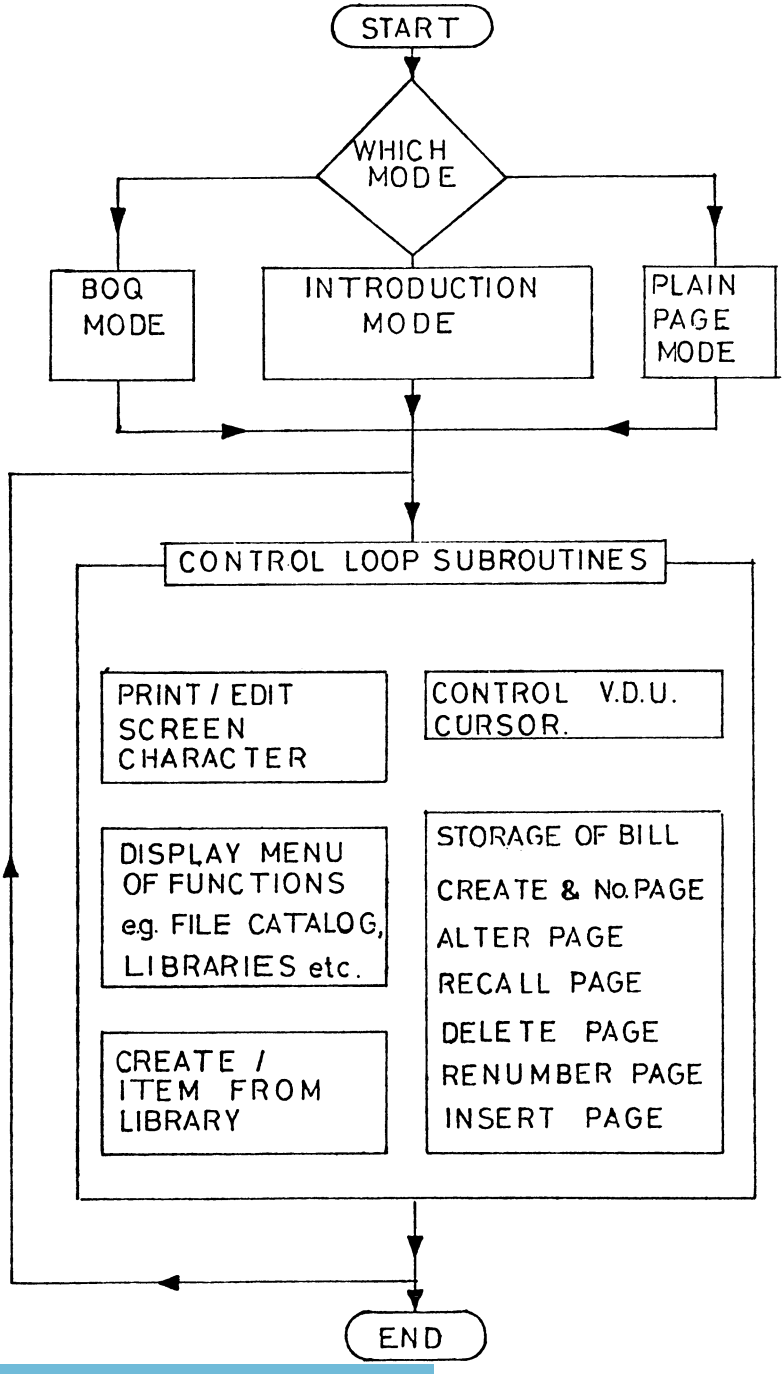
4 At the first stage - computing the quantities from drawings and handwriting the first draft of the text - the engineer can (with the assistance of the computer and on the basis of his/her experience) assemble a number of pages of "likely" items on paper copy output; retire to the drawing board; compute the necessary quantities; identify and include missing items; delete unused items; return to the computer and edit the "likely-items" list to a final draft.

ACKNOWLEDGEMENTS

The first two authors wish to thank the Department of Education for Northern Ireland and Kirk, McClure and Morton (Consulting Engineers) for the financial support given to this project. The authors also wish to thank Dr. G. Stevenson and Mr. R. McCombe, DOE(NI) Roads, and Mr. R. Weedon, Hewlett Packard for assistance given.

REFERENCES

1 Barnes, N.M.L., "Measurement in Contract Control", 1977.
2 Hewlett-Packard, "System 45 Operating and Programming", 1979.
3 Institution of Civil Engineers, "Civil Engineering Standard Method of Measurement", 1976.
4 Institution of Civil Engineers, "Standard Method of Measurement for Civil Engineering Quantities", 1955.
5. Institution of Civil Engineers and Federation of Civil Engineering Contractors, "The ICE Conditions of Contract", 1979.



PROGRAM FLOW CHART FIG 1.

THE VARIATIONAL INEQUALITY METHOD ON CONTACT PROBLEMS AND ITS APPLICATION SOFTWARE

Cui Jun-Zhi
The Computing Center, Academia Sinica, China.
Li Guo-Ren
Chengdu University of Science and Technology, China.
Li Guang-Zhong
Institute of Chengdu Hydroelectric Power Survey and Design, China.
Liang Fu-Gang, Haung Yu-Xia, Shi Guang-Jue
The Computing Center, Academia Sinica, China.

Abstract

In this paper the variational inequality theory and numerical method on contact problems are briefly described first, and then the structure of its application software is outlined. Finally the numerical results of its application to gravity dam with vertical gaps are given out. It shows that both of the numerical method and its software are valid.

THE THEORY AND THE METHOD

The formulation of variational inequality

The problems of the stress analysis of contact mutually between bodies are often encountered in many engineering, such as massive concrete dams, mechanical manufacture··· and so on. They all belong to the type of non-Hertz contact problems. If there exist friction on contact surface between bodies the

superposition principle can not be applied to them even though the bodies all are elastic. The stresses caused by a certain loads do not only depend on the amount of the loads, but also depend on the way of applying loads. So we must formulate them in incremental theory. Below we only consider the elastic contact problem with initial gaps in two dimension case. The essential suppositions are that the deformations are very small and the loads are applied in increment form step by step.

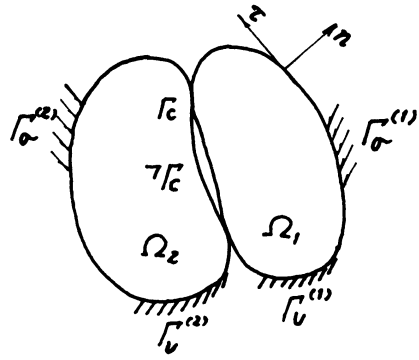


Fig. 1. The contact model .

Now we consider the elastic contact system shown in Fig.1. The elastic contact problem in increment form can be described as follows [2] (P_0): Suppose that, at a certain time, the contact system is in equilibrium, the contact boundary and the stresses being Γ^*, $\sigma^*=(\sigma_x^*, \sigma_y^*, \sigma_{xy}^*)^T$ respectively, and that at this time a new group of loads is applied to the system, i.e

$$f=\begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \text{ in } \Omega; \quad \bar{p}=\begin{pmatrix} \bar{p}_1 \\ \bar{p}_2 \end{pmatrix}, \text{ on } \Gamma_\sigma; \quad \bar{u}=\begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix}, \text{ on } \Gamma_u . \qquad (1)$$

Then what changes of stresses and contact boundary will occur? Denote the increments of the displacements and stresses by u and σ respectively, and the total stresses by $\bar{\sigma} = \sigma^* + \sigma$. Denote the contact boundary by Γ_c, and the non-contact boundary by $\neg\Gamma_c$ for the time being. Let $\Gamma_a = \Gamma_c + \neg\Gamma_c$.

In [3] we have already discussed that there is a group of unified nonlinear contact conditions on Γ_a , which is

$$u_n^{(2)} - u_n^{(1)} + \delta^* \geq 0 , \qquad \sigma_n^* + \sigma_n \leq 0 \tag{2}$$

$$(u_n^{(2)} - u_n^{(1)} + \delta^*)(\sigma_n^* + \sigma_n) = 0 \tag{3}$$

$$\text{sign}(\sigma_\tau^* + \sigma_\tau)(\sigma_\tau^* + \sigma_\tau) + F(\sigma_n^* + \sigma_n) \leq 0 \tag{4}$$

$$\text{sign}(\sigma_\tau^* + \sigma_\tau)(u_\tau^{(2)} - u_\tau^{(1)}) \geq 0 \tag{5}$$

$$[\text{sign}(\sigma^*+\sigma_\tau)F(\sigma_n^*+\sigma_n)+(\sigma_\tau^*+\sigma_\tau)](u_\tau^{(2)}-u_\tau^{(1)})=0 \tag{6}$$

where δ^* represents the width of gaps between bodies at the time t, $n=\{n_i\}$ and $\tau=\{\tau_i\}$ represent outward normal and tangent; the subscripts denote the components corresponding the directions, and the superscripts the quantities in corresponding bodies; and F is the coefficient of friction.

(1)-(6) combine with the Lame equations of elastic mechanics to form an elliptic problem with the nonlinear boundary conditions (P_1) [3]. It is too difficult to solve it directly. So we turn to the variational inequality form equivalent to (P_1). Below we will briefly present it.

The variational inequality form of the problem (P_0) is that (P_2):

to find u such that

$$a(u, v-u)+j(u,v)-j(u,u) \geq L(v-u), \quad \forall\, v \in U_{ad} \tag{7}$$

where

$$U_{ad}=\left\{ v \;\middle|\; \begin{array}{l} v=\{v_i\}\ ,\ v_i \in H^1(\Omega),\ v_i=0,\ \text{on}\ \ \Gamma_u \\ (v_i^{(2)}-v_i^{(1)})\cdot n_i+\delta^* \geq 0,\ \text{on}\ \Gamma_a \end{array} \right\} \tag{8}$$

$$a(u,v)= \int_\Omega a_{ijkh}\varepsilon_{kh}(u)\varepsilon_{ij}(v)\, dx \tag{9}$$

$$j(u,v)= \int_{\Gamma_a} |\sigma_n^*+\sigma_n(u)|\cdot|v_\tau^{(1)}-v_\tau^{(2)}|\, d\Gamma \tag{10}$$

$$L(v)= \int_\Omega f_i v_i dx+\int_{\Gamma_\sigma} \bar{p}_i v_i d\Gamma+\int_{\Gamma_a} \{\sigma_\tau^*(v_\tau^{(2)}-v_\tau^{(1)}) + \sigma_n^*\,(v_n^{(2)}-v_n^{(1)})\}d\Gamma \tag{11}$$

where $[a_{ijkh}]$ represents the elastic matrix; $j(u,v)$ is the work used up by the friction force, which is caused by u when the slip occurs; the third term of the right hand side of (11) is the work exerted by the force σ^* on the contact surfaces just then.

When the contact surfaces are frictionless, i.e $F = 0$,

$$\sigma_\tau^* = \sigma_\tau = 0 \; ; \qquad j(u,v) = 0 \tag{12}$$

$$L(v) = \int_\Omega f_i v_i dx + \int_{\Gamma_\sigma} p_i v_i d\Gamma + \int_{\Gamma_c} \sigma_n^* (v_n^{(2)} - v_n^{(1)}) d\Gamma \; . \tag{13}$$

Then (P_2) becomes that (P_3):

to find u such that

$$a(u, v-u) \geq L(v-u), \qquad \forall\, v \in U_{ad} \tag{14}$$

we can also prove that (P_3) is equivalent to (P_3'):

$$\text{Min} \quad J(v) = \frac{1}{2} a(v,v) - L(v) \qquad v \in U_{ad} \tag{15}$$

It is easy to prove that (P_3) has a unique solution, yet it can not be proved that (P_2) has a unique solution.

An iterative scheme of the solution of P_2

Since the work used up by the friction force is unknown before we work out u, it is too difficult also to solve (P_2) directly. Below we introduce an iterative scheme. Using it we have successfully solved the numerical solution of (P_2) in the practice engineering.

First we notice that for arbitrary u^* such that $\sigma(u^*)|_{\Gamma_a} \in L^2(\Gamma_a)$, the following problem (P_2') of the variational inequality has a unique solution u,

(P_2'): to find u such that

$$a(u,v-u)+j(u^*,v)-j(u^*,u) \geq L(v-u), \quad \forall\, v \in U_{ad} \tag{16}$$

Therefore starting from arbitrary $u^0 \in U_{ad} \cap \{ v \mid \sigma(v)|_{\Gamma_a} \in L^2(\Gamma_a)\}$, the iterative scheme can be constructed as follows

to find u^{n+1} such that

$$a(u^{n+1},v-u^{n+1})+j(u^n,v)-j(u^n,u^{n+1}) \geq L(v-u^{n+1}),$$
$$\forall\, v \in U_{ad}\ ,\quad n = 0,1,2,\cdots \tag{17}$$

We can prove that the iterative scheme (17) is equivalent to the following

for arbitrary u^0 to find u^{n+1} step by step such that

$$J(u^n,u^{n+1}) = \underset{v \in U_{ad}}{\text{Min}}\ J(u^n, v)\ ,\quad n = 0,1,2,\cdots \tag{18}$$

$$J(u^n, v) = \frac{1}{2}a(v,v)+j(u^n,v)-L(v) \tag{19}$$

Up to now, we can not demostrate the convergence of the sequence $\{u^{n+1}\}$. But in practice applications if Mes (Γ_c) is small enough, $\{u^n\}$ is convergent. We can take u^n, n being large enough, as an approximate solution of P_2. We should notice that Γ_c is the part of Γ_a where the slip possibly occurs. A way of making Mes (Γ_c) be small enough is reducing increments f and $\bar{P}$ of loads.

The numerical method

First we discuss the method of solving the problem (P_3). We divide the domain Ω into the set of triangles, shown in Fig.4.b, and an approximate domain Ω^h. In each triangle a linear approximation to the displacements is constructed. In this way, for each $v \in U_{ad}$, we can get an approximate v^h. Let

$$U^h_{ad} = \{v^h \mid v \in U_{ad} \cap (C^0(\Omega))^2,\ v^h \ \text{approximates}\ v\ \} \tag{20}$$

By calculating integration on U^h_{ad} (P_3) is changed into P^h_3 :

$$\text{Min}\ \ J_h(v^h) = \frac{1}{2}\,(Kv,\ v) - (f,v)$$
$$BV \geq 0 \tag{21}$$

where $BV \geq 0$ is put up by approaching restrained condition

$(v_i^{(2)}-v_i^{(1)})\cdot n_i + \delta^* \geq 0$ on Γ_a, and B is a m × N sparse matrix, whose rank equals m; and $v=(v_1, v_2,\cdots, v_N)^T$ is the vector of the V's nodal value. We should notice that in (21) the prescribed constraints $v_i = \bar{u}_i$ on Γ_u have been

eliminated, so $K = \{K_{ij}\}$ is a symmetric and possitively definite matrix. We should also notice that $Bv \geq 0$ can be divided into several restrained subgroups $B_i v \geq 0$ $(i=1,2,\cdots,t)$, which have no relation in between. Each $B_i v \geq 0$ of the subgroups has a less order m_i, and $\sum_{i=1}^{t} m_i = m$. And then reasonably expanding B_i into $\bar{B}_i$ we can construct a local mapping of unknowns from v_i to u_i for each subgroup

$$u_i' = \bar{B}_i v_i$$

where v_i is the part of unknown involved by $B_i v \geq 0$. By the local mappings the problem (21) can be changed into the following form

$$\begin{aligned} &\text{Min} \quad J(u) = \frac{1}{2}(K'u,u) - (f',u) \\ &\qquad u_i \geq 0, \quad i=1,2,\cdots, m . \end{aligned} \tag{22}$$

For such optimization problem (22) we can use the successive relaxation method to solve it. The iterative scheme is the following

$$\begin{aligned} r_i^{(K+1)} &= f_i' - \sum_{j<i} K'_{ij} u_j^{(K+1)} - \sum_{j\geq i} K'_{ij} u_j^{(K)} \\ u_i^{(k+1)} &= \begin{cases} \max\{0,\ u_i^{(K)} + \omega\, r_i^{(K+1)}/K'_{ii}\} & \text{as } i \leq m \\ u_i^{(K+1)} + \omega\, r_i^{(K+1)}/K'_{ii}, & \text{as } i > m \end{cases} \\ &\qquad k = 0,\ 1,\ 2,\ \cdots , \end{aligned} \tag{23}$$

where ω is called as relaxation factor. In our computation experience ω can be chosen between 1.8125 to 1.9375 for the contact problem of bodies.

Using the theorem in [1], it can be proved that the sequence $\{u^k\}$ generated by the scheme (23) converges to the true solution u of the problem (22).

Now we turn to discussing the numerical method of solving friction problem (18). By the same discretization procedure previously we can obtain the discrete form of the problem (18) as follows

$$\text{Min}\quad J_h(u^n,v^h) = \frac{1}{2}(Kv,v) + \Sigma|(c_{ij},v_j)| - (f,v)$$
$$BV \geq 0 \qquad (24)$$

where $\sum_i |(c_{ij},v_j)|$ is set up by calculating $j(u^n,v)$, and $c=[c_{ij}]$ is a $P \times N$ sparse matrix whose rank is equal to P. Because BV and CV come from discretizations of the normal restraints and the tangential slip on Γ_a respectively, BV and CV are independent to each other. So the matrix $\begin{pmatrix} B \\ C \end{pmatrix}$ has the rank p+m. Evidently, p+m<<N.

Now transforming CV and BV by previous procedure of local mapping at the same time we can alter (24) into the following

$$\text{Min } J(u) = \frac{1}{2}(K' u,u) + \sum_{m+1\leq i\leq p+m} |u_i| - (f',u) \qquad (25)$$
$$u_i \geq 0,\ i = 1,2,\cdots,m$$

In the practice computing, according to σ^*, which have been known, or the iterative value at a time, we can experimentally asserted the sign of tangential dislocations $(u_\tau^{(2)}-u_\tau^{(1)})_i$ appearing on the gaps under the new incremental loads. And then we can make out the sign of u(i=m+1, m+2, ···, m+p). So we can still use the SOR scheme (23) to work out the results u^{n+1} that we want.

THE COMPUTER PROGRAM

The aforesaid method on the elastic contact problem with friction can combine with the method of the elastoplasticity analysis formulated in incremental form or the method of the creep analysis under various loads including heat to analyse various engineering problems. Below we sum up the frame of the computer program on them.

The leading flowchart of the program is shown in Fig.2.

Remark 1. The program adopts a kind of semiautomatic approach of generating meshes. The structure is divided into a few large zones first, shown in Fig.4.a, and the fineness of element subdivision within each is specified, and the local fineness round some prescribed points previously is also given. The

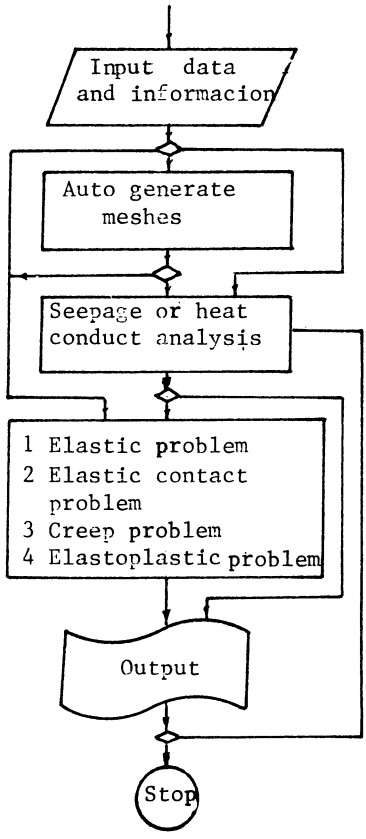


Fig.2. The leading flowchart.

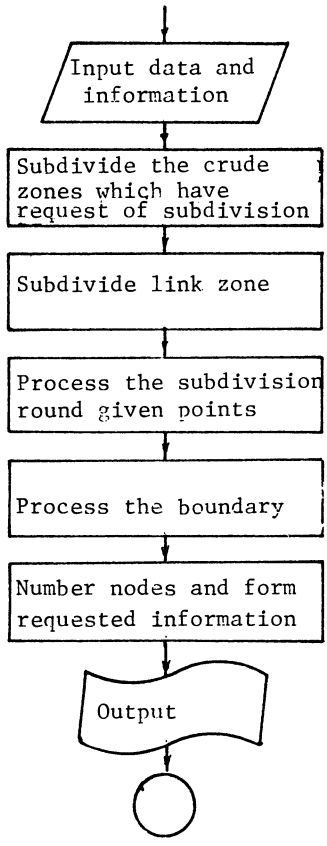
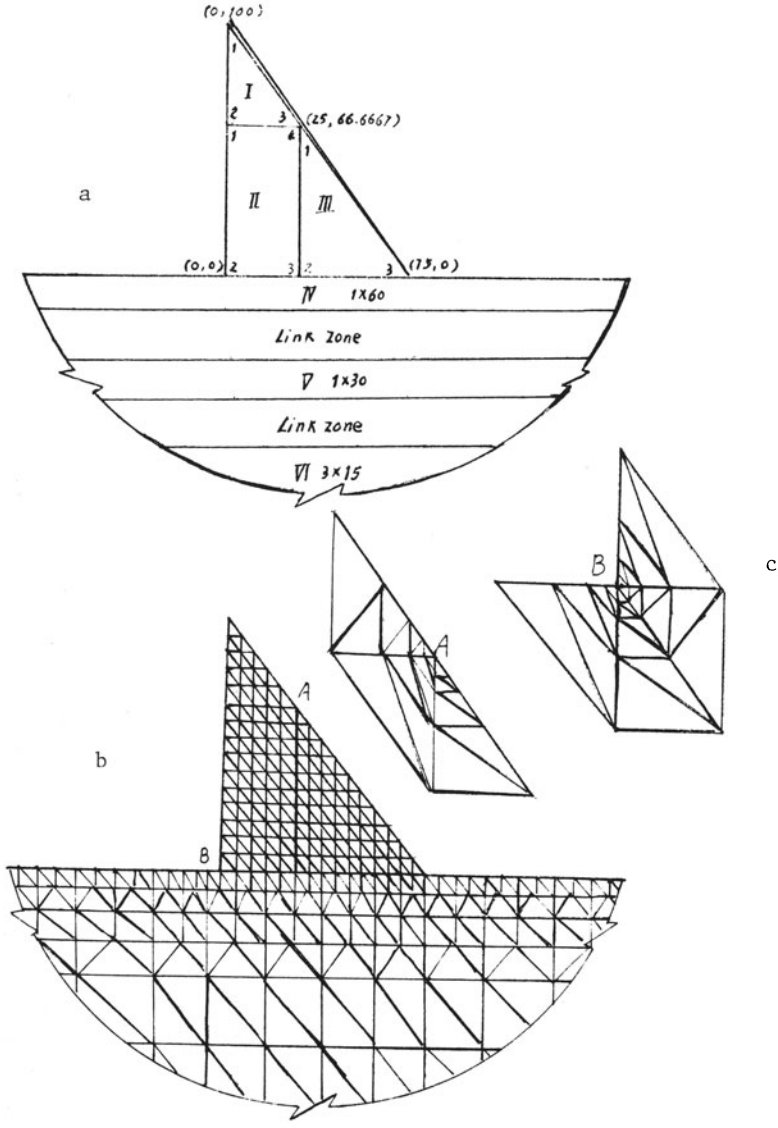


Fig.3. The flowchart of automatic subdivision

initial data about crude division is input in the normal way and the subdivision proceeds automatically. The flowchart of this subprogram is shown in Fig.3. The result of subdivision for Fig.4.a is shown in Fig.4.b.

Remark 2, In order to be able to analyse more problems of practical engireering, the subprogram of elastic contact analysis also has some special functions satisfying analysis requirements of some special engineering. It can auto-combine with heat or seep subprogram, and also auto-treat many loads,



a) The crude division of gravity dam with gaps.

Fig.4, b) The result of subdivision.

c) Sundivision round the prescribed points.

for the concrete gravity dam with vertical gaps, for example, it can automatically deal with the volume force, water force on the outside, seepage pressure, heat load, friction on the internal surfaces in between and sticking force if the internal surfaces have been grouted.

Remark 3. The method of creep analysis is presented in [11]. The subprogram can deal with the problem of variable domain, the problem of the variation of elastic modulus with time and the influence of historical stresses to the stresses at present. So the construction process of dam can be simulated with the computed region increasing in accordance with the progress of construction.

THE PRACTICE APPLICATION

The previous methods and their program have been applied to analysing various problems, including contact model problem with experiment results. Here we only illustrate one of them, which has measurement results that the computing results can compare with. The object analysed is a practical gravity dam. The ninth dam section, selected for study, has two gaps of vertical open joints, shown in Fig.5. In 1972, within 10 days, the reservoir level of water was raised from 493.0 m to 502.0 m, 511.0 m and then 520.0 m. Meanwhile, measurements were made on the horizontal displacements of dam crest, openings of longitudinal joints and strains of embedded gauges. Taking the deformation and stresses of reservoir level 493.0 m as initial conditions and applying incremental water loads in three steps, computation of the deformations and stresses were made. In computation, the modulus of elasticity of concrete was taken as $3\times10^5 kg/cm^2$, that of rock foundation, $1.5\times10^5 kg/cm^2$, and poisson's ratio of concrete, 0.167 and that of rock, 0.25. The meshes have 1620 triangles. The comparison between computer results and measured data have been shown in Fig.6-9 . They are very close.

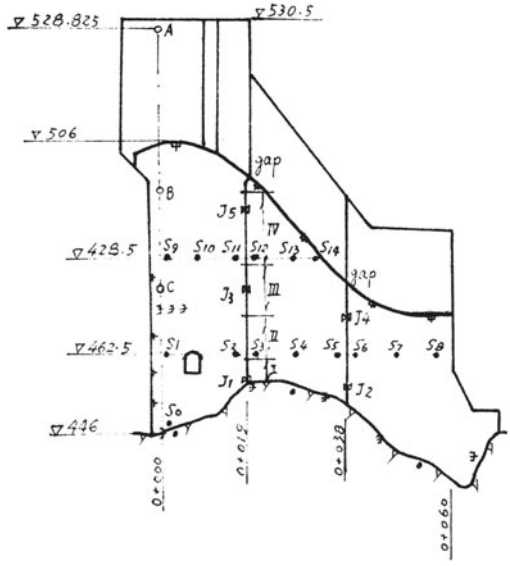


Fig.5. 9th dam section and arrangement of observation instruments.

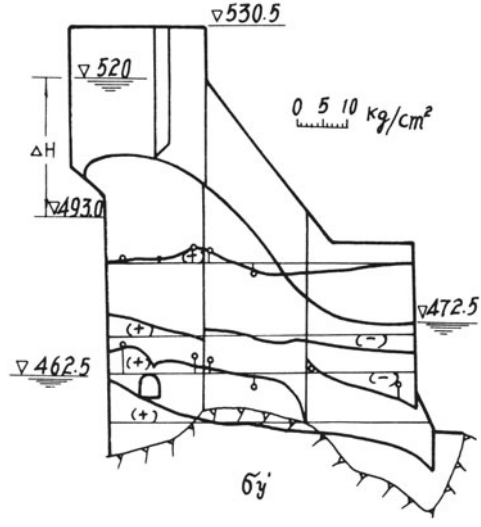


Fig.6. Distribution of σ_y increment from 493.0 m to 520.0 m

o observed ----- calculated .

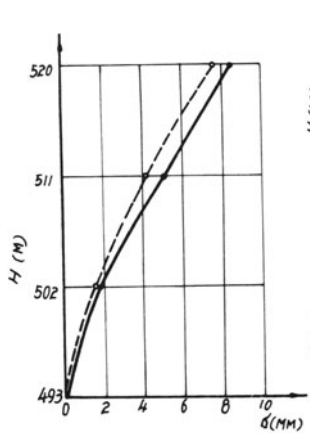


Fig.7. Horizontal displacement of dam crest

_____ observed

--------calculated

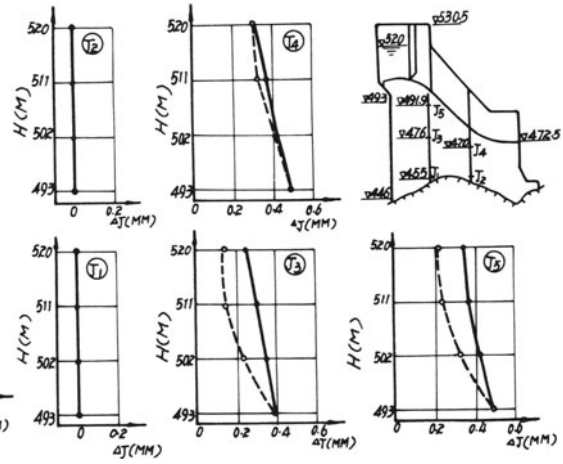


Fig.8. Variation of gap width at some joint points.

_____ observed , ----- calculated.

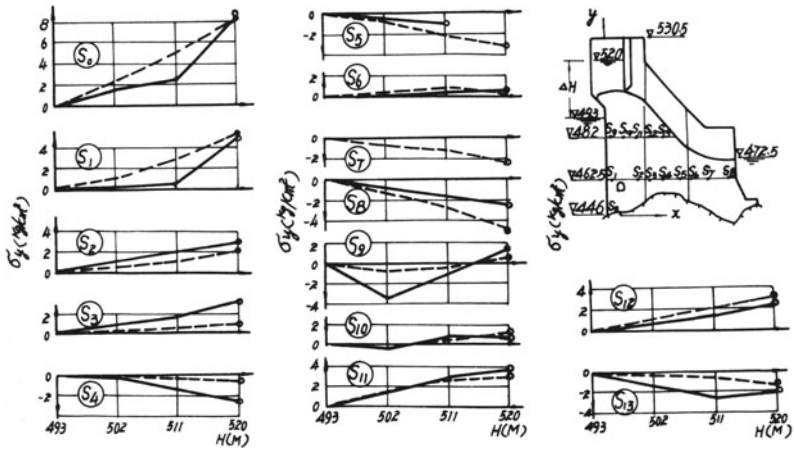


Fig. 9 . Vertical stress increments σ_y .

_____ observed , ----- Calculated .

CONCLUSION

Through computing analysis of model experiment or engineering practice, the reliability of previous method and its program was affirmed at the same time. The practice of computation has shown that they are powerful.

REFERENCES

Cea J. "Lectures on optimization, Theory and algorithms", Tata Institute of Fundamental Research, Bombay, Berlin, 1978.

Cui Jun-Zhi, Li Guang-Zong, Liang Fu-Gang, Shi Guang-Jue and Li Guo-Ren, "On the method of analysis of elastic contact problem", "Transactions of the meeting of computer application in water conservancy and hydroelectric power engineering, 1974, or "The finite element methods and their applications", The Institute of Computer Technology of Academia Sinica, 1975.

Cui Jun-Zhi, "On the problems of elastic contact with initial gaps", Acta Mechanica Sinica No.3, 1980, pp268-277.

Cui Jun-Zhi, "The solution and iterative scheme of the elastic contact problems with initial gaps", Transaction of National Conference on Computational Mechanics, Hang Chow, China, Nov. 1980, pp33-38.

Cui Jun-Zhi, Huang Yu-Xia, Liang Fu-Gang, Shi Guang-Jue, "The solution and solving method of the elastic contact problem with initial gaps", China-France Symposium on Finite Element Method, Beijing, China. 4. 1982.

Cui Jun-Zhi, Haung Yu-Xia, Shi Guang-Jue and Liang Fu-Gang, "The program of finite element method of two dimension problem, User manual", Research report, 1978.

Duvaut G. and Lions J.L. "Les inequations en mechanique et en

physique", Donod, Paris, 1972.

Glowinski R. "Numerical method for non-lineor variational problem", Nov. 1979.

515 Research Group, "The interface connection in a gravity dam with open longitudinal joint and its operation state", Research report, 1978.

515 Scientific Research Group, "Investigation of the feasibility of a gravity dam put into operation at higher reservoir water level," Research report, 1978.

515 Scientific Research Group, "The finite element method on plane creep problem", Math. Numer. Sinica, 5, 1978.

515 Scientific Research Group., "The finite element method on heat conduct analysis", Reseach report. 1978.

515 Scientific Research Group, "The finite element method of the transient heat conduction equation and the principle of maximun norm", Math. Numer, Sinica, 5, 1982.

USE OF A MICROCOMPUTER TO DETERMINE SLIP-LINE FIELDS FOR STRIP FOOTINGS

* J. H. Atkinson, + R. Delpak, @ G. T. Jones.

* Reader in Soil Mechanics, City Univ., London.
+ Principal Lecturer in Civil Eng., Poly of Wales.
@ Research student and Snr. Progr., Poly of Wales.

ABSTRACT

This paper describes a computer program written in the FORTRAN language,to determine slip-line fields by using the "associated fields" method, for drained and undrained loading conditions in plane strain. The program employs finite difference approximations and iteration, to obtain a stress field and a set of displacements according to the boundary conditions. In addition, values are calculated for work dissipated internally and externally for each chosen collapse mode. Selected results appear in the tables which are given at the end of the text.

INTRODUCTION

The theory of soil plasticity has been developed over a period of many years, and results mainly from investigations into the load bearing behavior of soils and of other materials. The presence of slip-planes (or slip-lines, for plane strain), that is planes along which failure of soil masses occurs, was noted by the earliest investigators. Methods based on the positions of slip-planes are well established and continue to provide useful results. The literature is extensive and no attempt will be made to give a detailed treatment here. However, an outline which indicates the most notable developments is presented.

The earliest theoretical treatment of soil strength was due to Coulomb (1773), who proposed the well known failure criterion which bears his name. Rankine (1857) investigated limiting plastic equilibrium and proposed his earth pressure theories for retaining walls. Saint-Venant (1870) described an ideal plastic material, for which three-dimensional relationships between stress and rate of plastic

strain were subsequently suggested by Levy (1873). The method of characteristics was first used by Massau (1899) for the approximate determination of stress fields, particularly for the solution of plane strain problems for cohesionless soils and a setof differential equations defining the geometry of a set of characteristics was obtained by Kotter(1903), in terms of curvilinear coordinates. The loads required to indent plane surfaces and wedges by a flat die were calculated by Prandtl (1921), who gave possibly the first practical solutions using a simplified slip-line field. Subsequent work by Fellenius (1926) also employed various simple shapes for slip-lines, and was further developed by other investigators, the most notable being Terzaghi (1943) whose published works in soil mechanics are well known. The development of numerical methods is based largely on the work of Sokolovskii (1965), who used finite differences as suggested by Hill (1950) in his classic theoretical treatise. A limitation of classical plasticity theory is due to the differences between the deformation behaviour of ideal and real materials, and for this reason much of the work in soil mechanics in recent years has been concentrated on the actual stress-strain behaviour of soils. This work has led to the "critical state" model, first described by Schofield and Wroth (1968), which combines the relevant theories of elasticity, plasticity and friction, and is the foundation of a consistent theory of soil plasticity. An important feature of the critical state model is that, at failure,soils reach certain well defined critical or ultimate states, at which plastic shear straining continues without further change of stress, pore pressure or volume. In the treatment of the stability of foundations described in this paper, the soil is taken to be at a critical state of failure and this requirement prescribes the states of stress and plastic strain in the soil, and particularly on the slip planes through the soil.

NOTATION

The notation used in the text and diagrams is that which is commonly used in soil mechanics, and is briefly described as follows.

x, z - Cartesian axes in plane strain.
σ_x, σ_z - normal stresses referred to x and z axes.
σ_1, σ_3 - principal stresses (major and minor).
τ - shear strain.
$\delta\epsilon$ - increment of normal strain.
$\delta\gamma$ - increment of shear strain.
s, t - $\frac{1}{2}$(sum) and $\frac{1}{2}$(diff.) principal total stresses.

s', t' - (sum) and (diff.) principal efftv.stresses.
γ - unit soil weight.
c_u - coefficient of undrained cohesion.
ϕ_{cs} - critical state angle of friction.
ψ - angle of dilation.
η - angle made by σ_1 with z-axis.
α, β - characteristics, also called slip-lines.
// - lines drawn thus indicate slip-lines.

It is assumed throughout that soil is homogeneous, isotropic and perfectly plastic. Also that the normality condition is satisfied and the flow-rule is associated. Plane strain conditions are used, so that the analysis is effectively simplified to that of two dimensions only.

THE STRESS FIELD FOR UNDRAINED LOADING

For a region which is in a state of undrained failure everywhere, the stresses satisfy both the conditions for equilibrium and the failure criterion.

The equilibrium conditions in terms of total stresses are given by:

$$\frac{\partial \sigma_z}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} = \gamma \qquad (1)$$

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{zx}}{\partial z} = 0 \qquad (2)$$

and the failure criterion for undrained loading of saturated soil in plane strain is given by:

$$c_u = t = \tfrac{1}{2}(\sigma_1 - \sigma_3) \qquad (3)$$

where c_u is the undrained shear strength.
As the failure envelope for undrained loading is horizontal, the state of stress is completely determined from the position of the centre of the Mohr's circle, and it is convenient to define this by the "mean compressive stress" s, given as

$$s = \tfrac{1}{2}(\sigma_1 + \sigma_3) \qquad (4)$$

The following relationships are then obtained for the stresses.

$$\sigma_x = s + c_u \cos(180^\circ - 2\eta) = s - c_u \cos 2\eta \qquad (5)$$
$$\sigma_z = s - c_u \cos(180^\circ - 2\eta) = s + c_u \cos 2\eta \qquad (6)$$
$$\tau_{zx} = \tau_{xz} = c_u \sin(180^\circ - 2\eta) = c_u \sin 2\eta \qquad (7)$$

After partial differentiation of Eqns. (5) to (7) with respect to x and z, and substitution into Eqns.

(1) and (2), we obtain

$$\frac{\partial s}{\partial z} - 2c_u \sin 2\eta \frac{\partial \eta}{\partial z} + 2c_u \cos 2\eta \frac{\partial \eta}{\partial x} = \gamma \qquad (8)$$

$$\frac{\partial s}{\partial x} + 2c_u \cos 2\eta \frac{\partial \eta}{\partial z} + 2c_u \sin 2\eta \frac{\partial \eta}{\partial x} = 0 \qquad (9)$$

Eqns (8) and (9) are classed as simultaneous partial differential equations in s and η, and they are of the hyperbolic kind. The usual method of solving equations of this type is that of characteristics, described in standard texts such as Abbott (1966). The actual numerical scheme of solution is described in a subsequent section. It can be shown that the characteristics of Eqns. (8) & (9) are

$$\frac{dx}{dz} = \tan(\eta \mp 45°) \qquad (10)$$

and along these characteristics, again in terms of total differentials

$$ds = \pm 2c_u d\eta + \gamma dz \qquad (11)$$

By using the Mohr's circle construction, it is found that the positions of failure planes which result from the analysis of stresses, coincide with those given by the above equations of the characteristics.

THE STRESS FIELD FOR DRAINED LOADING

For a region which is in a state of drained failure everywhere, the stresses satisfy both the conditions for equilibrium and the failure criterion.

The equilibrium conditions for drained failure, in terms of effective stresses are

$$\frac{\partial \sigma'_z}{\partial z} + \frac{\partial \tau'_{xz}}{\partial x} = \gamma - \frac{\partial u}{\partial z} \qquad (12)$$

$$\frac{\partial \sigma'_x}{\partial x} + \frac{\partial \tau'_{xz}}{\partial z} = - \frac{\partial u}{\partial x} \qquad (13)$$

where u is the pore pressure and, for dry soil u = 0. The failure criterion for drained loading of soil in plane strain is

$$t' = s' \sin \phi'_{cs} \qquad (14)$$

where ϕ'_{cs} represents the "critical state" value of the angle of internal friction.

The following relationships are then obtained for the stresses.

$$\sigma'_x = s' + t' \cos(180° - 2\eta) = s'(1 - \sin \phi'_{cs} \cos 2\eta) \qquad (15)$$

$$\sigma'_z = s' - t' \cos(180° - 2\eta) = s'(1 + \sin \phi'_{cs} \cos 2\eta) \qquad (16)$$

$$\tau'_{xz} = \tau'_{zx} = t' \sin(180° - 2\eta) = s' \sin\phi'_{cs} \sin 2\eta \qquad (17)$$

After partial differentiation of Eqns. (15) to (17) with respect to x and z, and substitution into Eqns. (12) and (13),

$$(1 + \sin\phi'_{cs}\cos 2\eta)\frac{\partial s'}{\partial z} + \sin\phi'_{cs}\sin 2\eta \frac{\partial s'}{\partial x} - 2s' \sin\phi'_{cs}\sin 2\eta \frac{\partial \eta}{\partial z} + 2s' \sin\phi'_{cs}\cos 2\eta \frac{\partial \eta}{\partial x} = \gamma - \frac{\partial u}{\partial z} \qquad (18)$$

$$\sin\phi'_{cs}\sin 2\eta \frac{\partial s'}{\partial z} + (1 - \sin\phi'_{cs}\cos 2\eta)\frac{\partial s'}{\partial x} + 2s' \sin\phi'_{cs}\cos 2\eta \frac{\partial \eta}{\partial z} + 2s' \sin\phi'_{cs}\sin 2\eta \frac{\partial \eta}{\partial x} = -\frac{\partial u}{\partial x} \qquad (19)$$

Equations (18) and (19) are a set of simultaneous partial differential equations in s and η, and again are of the hyperbolic kind. For these the characteristics can be shown to be

$$\frac{dx}{dz} = \tan[\eta \mp (45° - \tfrac{1}{2}\phi'_{cs})] \qquad (20)$$

Along these characteristics, total differentials are

$$ds' = \pm 2s' \tan\phi'_{cs}\, d\eta + \gamma(dz \pm \tan\phi'_{cs}\, dx) \qquad (21)$$

The Mohr's circle for stresses, shows that the positions of the failure planes coincide with those given by the above equations of the characteristics.The method of solving Eqns. (18) & (19) numerically along these characteristics is discussed as follows.

NUMERICAL METHOD OF SOLUTION ALONG CHARACTERISTICS

The significance of the characteristics of hyperbolic equations is that along these lines, total rather than partial differential relationships are obtained. The calculations commence at a boundary where values are known, and proceed along the characteristics. Since the hyperbolic equations which were obtained in the previous section are paired, then there will be two intersecting families of characteristics, for each of the drained and undrained loading cases. There are three distinct types of numerical problem associated with calculations along characteristics, these will be denoted by the letters I, C and M as in Figure 1, and are described separately in the paragraphs which follow.

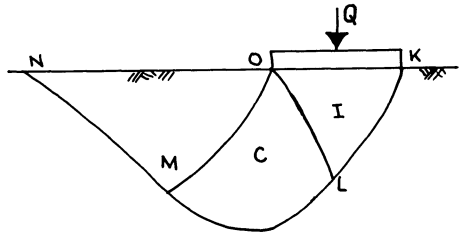


Figure 1. Failure region below a strip footing.

Problem I, The initial value or Cauchy problem.

In Figure 2, OK is a boundary (not a characteristic) along which initial values are known. Usually these valuesare normal and shear stresses along this boundary.

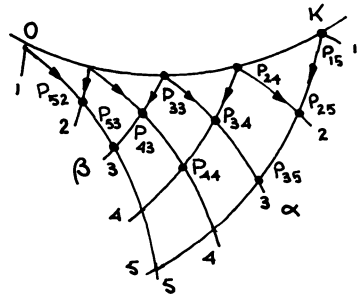


Figure 2. The Initial Value Problem.

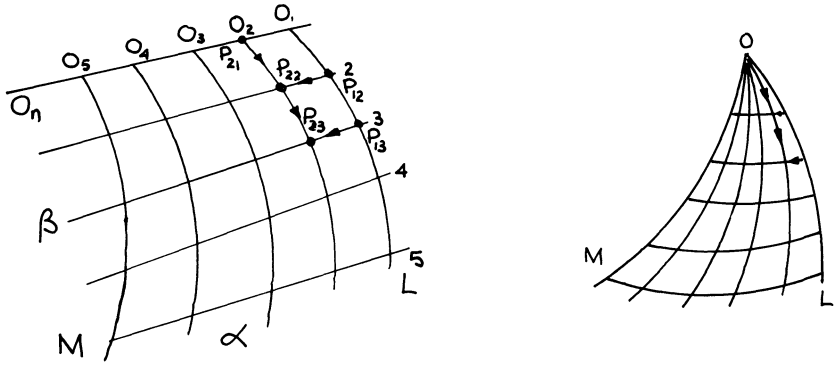
If the individual characteristics of the α and β families are labelled 1, 2, 3 .... as shown in Figure 2, then the point of intersection of an α line i witha β line j is denoted by P_{ij} . The calculations are then carried out as follows.

1. Values of x, z, s, η at P_{25} can be obtained from the known values at P_{15} and P_{24} .
2. Values of x, z, s, η at P_{34} can be obtained from the known values at P_{24} and P_{33} .
3. Continuing this process will determine all values along the row of points below the boundary OK.

Repeating the process in steps 1 to 3 for subsequent rows, progressively determines the required values at all points in the region OKL.
N.B. If the arc OK tends towards a characteristic, then the zone of intersections tends to vanish. Thus a solution cannot be continued across a characteris-

tic line without further information.

Problem C, The characteristic or Goursat problem.



(a) (b)

Figure 3. The Goursat problem.

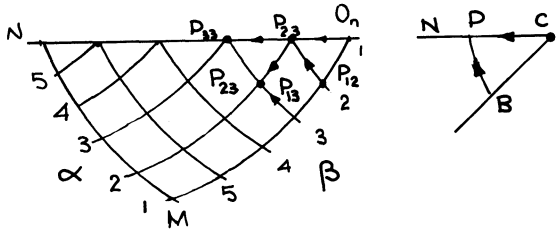
In Figure 3(a), O L is an α line. The points $O_1 \ldots O_n$ are infinitely close together on a β line; these points actually coincide, but to overcome the difficulty that O is point of stress discontinuity, values of η and s are specified at convenient intervals at the points $O_1 \ldots O_n$. The points of intersection in the mesh are numbered as for the previous problem (I).

The calculations proceed along the characteristics as follows.

1. Values of x, z, s, η at P_{12} and at P_{21} are used to calculate the values at P_{22} .
2. Values of x, z, s, η at P_{13} and at P_{22} are used to calculate the values at P_{23} .
3. Continuing the process will determine all required values along α line 2.

Repeating the process in steps 1 to 3 for subsequent rows (α lines), will determine the values at all points of intersection throughout the region OLM.

Problem M, The mixed problem.



(a) (b) (detail)

Figure 4. The mixed problem.

In Fig.4, OM is an α line and ON a non-characteristic line along which certain conditions are prescribed. Typically, ON may be a known boundary which is stress free or shear free. If values of x, z, s and η are known at a point such as P_{12} on OM, then these values may be used to calculate approximate values at a point such as P_{22} on the non-characteristic ON. In Fig.4(b), the point B represents P_{12} and the points C and P represent O and P_{22} respectively. If η is known along ON, and the position of this line is known, then referring to Fig.4(b), the equation of ON is given by

$$\frac{z_P - z_C}{z_N - z_C} = \frac{x_P - x_C}{x_N - x_C} \qquad (22)$$

The equation of characteristic BP can be written as

$$z_P - z_B = \tan\left(\frac{\eta_B + \eta_P}{2} + \left(45^\circ - \frac{\phi}{2}\right)\right)(x_P - x_B) \qquad (23)$$

The calculations proceed as follows.

1. Solve Eqns. (22) and (23) to give x_P and z_P ; s_P can now also be obtained, so that all required values at P ie. P_{22} in Fig. 4(a), are known.
2. Calculate x, z, s and η at P_{23} from the known values at P_{22} and P_{13} , as described previously for problem C, since the lines from P_{22} and P_{13} are both characteristics.

The process described above is repeated until all required values are found in the region OMN, so that the values at the boundary ON are determined.

Summary.

These calculation methods are applicable to undrained loading when the governing hyperbolic equations are (7) and (8), and to drained loading when the the governing equations are (17) and (18). Thus it is, in theory, possible to calculate stresses in any region of failing soil from a few known boundaries.

DETERMINATION OF THE VELOCITY FIELD

The methods of solution described in the previous sections involved stresses only, and solutions satisfied conditions of equilibrium and failure. To include calculations for displacements it is necessary to satisfy conditions of compatibility, together with some stress-strain relationships for the soil. For plastic deformation at failure, increments of displacement may be interpreted as velocities. It is assumed that all deformations occur only within the region of failing soil. The strain behaviour of the soil is characterised by an angle of dilation ψ, which is assumed to have a constant value throughout the region. For the undrained loading case $\psi = 0$, while for drained loading the value of ψ will be either positive or negative, according to whether the soil is dilating or compressing.

If the components of the displacement δw of a point are δh and δv in the x and z directions respectively, then the increments of strain are

$$\delta\epsilon_x = -\frac{\partial(\delta h)}{\partial x} \qquad (24)$$

$$\delta\epsilon_z = -\frac{\partial(\delta v)}{\partial z} \qquad (25)$$

$$\delta\gamma_{xz} = \delta\gamma_{zx} = -\left(\frac{\partial(\delta v)}{\partial x} + \frac{\partial(\delta h)}{\partial z}\right) \qquad (26)$$

For compatibility, the following relationships can be derived from the geometry of the Mohr's circle.

$$(\delta\epsilon_x - \delta\epsilon_z) = \left(\tfrac{1}{2}\delta\gamma_{xz} + \tfrac{1}{2}\delta\gamma_{zx}\right)\cot(180^\circ - 2\eta) \qquad (27)$$

$$\tfrac{1}{2}(\delta\epsilon_x + \delta\epsilon_z) = -\tfrac{1}{2}\delta\gamma_{xz}\sin\psi\,\mathrm{cosec}(180^\circ - 2\eta) \qquad (28)$$

Substituting the various terms from Eqns.(24) to (26)

$$\frac{\partial(\delta h)}{\partial x} - \frac{\partial(\delta v)}{\partial z} + \cot 2\eta\left[\frac{\partial(\delta v)}{\partial x} + \frac{\partial(\delta h)}{\partial z}\right] = 0 \qquad (29)$$

$$\frac{\partial(\delta h)}{\partial x} + \frac{\partial(\delta v)}{\partial z} + \sin\psi \operatorname{cosec} 2\eta\left[\frac{\partial(\delta v)}{\partial x} + \frac{\partial(\delta h)}{\partial z}\right] = 0 \qquad (30)$$

Equations (29) and (30) are a pair of simultaneous hyperbolic partial differential equations, and can be solved using the methods described previously for the others of this kind, making use of characteristics.

The equations of the characteristics are

$$\frac{dx}{dz} = \tan\left[\eta \mp (45° - \tfrac{1}{2}\psi)\right] \qquad (31)$$

(the negative sign applies for the α line and the positive sign for the β line). It can be seen that Equation (31) defines two sets of characteristics; and they make equal angles of $(45 - \psi/2)$ on either side of the direction of $\delta\epsilon_1$, so that the angle between the characteristics is $(90 - \psi)$. A Mohr's circle shows that the directions obtained for the strain increment characteristics, coincide with those defined by Eqn. (31) above. The four equations (29) to (31) involve five unknowns (δv, δh, η , x and z), and so will not provide a unique solution unless some appropriate restraint is included. This additional restraint is given by the stress-stain properties of the soil. It is assumed that the directions of the principal stress and of the principal strain increment coincide, and also that the angles of dilation and friction are equal. Thus the angles η are the same for stress and velocity solutions, and $\psi = \phi'$ so that the characteristics of stress and of velocity coincide.

CALCULATIONS FOR WORK DONE BY EXTERNAL LOADS AND DISSIPATED BY PLASTIC SHEARING

The calculation of work done by the external loads involves summing the products of stress and velocity along the soil surface; this is done by means of a numerical integration technique. The work due to soil weight is also calculated.

The work done in the failure region due to plastic shearing, involves separate calculation of work due to deformation, and that due to sliding along the boundary with the rigid region. The work done as a result of deformation is found for each individual element of the slip-line mesh, ie. each curvilinear figure formed by the intersection of two pairs of α and β lines, as shown in Figure 5.

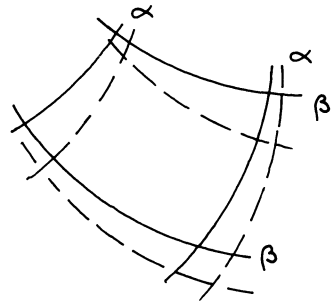


Figure 5. Deformation of an element.

In order to verify that stress and velocity fields obtained are valid, the following conditions are checked automatically.

(a) That the work dissipated everywhere is positive.
(b) That the work due to external forces is equal to the work done internally.

USE OF A MICRO COMPUTER

The finite difference methods described earlier for solving hyperbolic equations along characteristics, require a limited amount of computer storage and the iterations are found to converge fairly quickly. The methods are thus well suited to micro-computing.

Computer Storage Requirements.

The calculations require only core storage so that no intermediate output, for subsequent recall, is employed. The actual amount of storage necessary for the solution of a problem, will necessarily depend on the number of initial nodes specified, as this will determine the total number of nodes in the mesh.

The following table indicates the approximate storage requirements.

Table 1.

Storage requirements.

No. of Initial nodes	Total Nodes	No. of Values stored
5	45	1 k
7	91	2 k
9	153	3 k

(Several bytes are required to store one value)

Input Data.

The values required for input are as follows:
(a) c , ϕ , γ
(b) number of initial points.
(c) distance between initial points (constant).
(d) loading distribution.
(e) initial point velocity distribution.

Output Data.

The values available for output are as follows:
(a) bearing capacity.
(b) x, z coordinates; s; η at each node.
(c) velocity components at each node.
(d) work calculations for each slip-line element.
(e) work due to external loads.

SOLUTIONS FOR STRIP FOOTINGS

The characteristic method has been used to calculate stresses and displacements below a long strip footing placed at the surface of the soil, as illustrated in Figure 6.

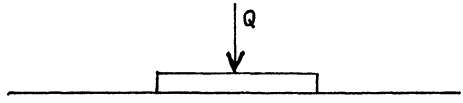


Figure 6. A surface footing.

The loading conditions must be specified below the footing in terms of stress and displacement, and a number of alternative distributions have been examined.

Loading Boundary Conditions.

The distribution of stresses at the initial, equally spaced points along the footing, is calculated from the loading which is specified as one of the following options:

1. Linear (constant, increasing or decreasing).
2. Parabolic.
3. Circular arc.
4. Individually prescribed at the initial points.

Some examples are shown in Figure 7.

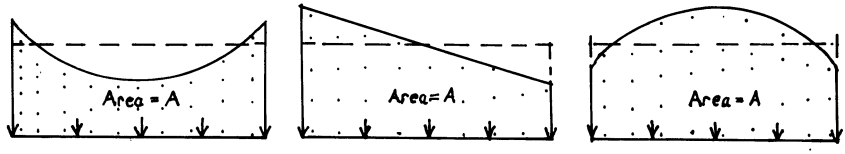


Figure 7. Specification of loading.

Velocity Boundary Conditions.

The velocities at the initial nodal points are specified by selecting one of the options provided; these are identical with theloading options, ie. linear, parabolic, circular arc or individually supplied. Selection of a non-linear velocity distribution provides a simulation of flexible footings, whilst a linear distribution simulates rigid footing behaviour.

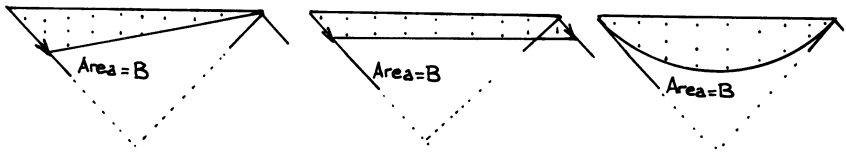


Figure 8. Specification of velocities.

Results for Bearing Capacity.

The bearing capacity figure is usually given as a function of the dimensionless soil weight parameter, G where $G = \gamma B/2c_u$.

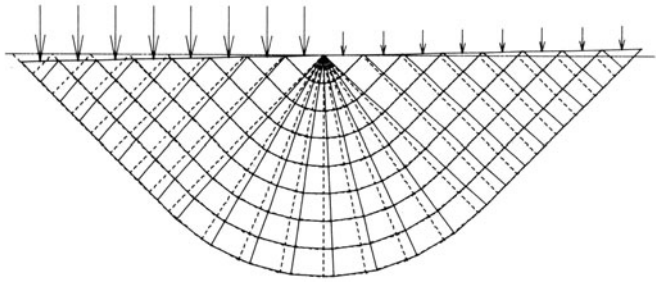
Table 2.

	Average Bearing Capacity (q/c_u)		
ϕ_{cs}	G = 0.1	G = 1	G = 10
0	5.14	5.14	5.14
10	8.35	9.01	13.30
20	14.97	17.64	37.10
30	30.55	42.27	125.32
40	81.0	137.0	568.0

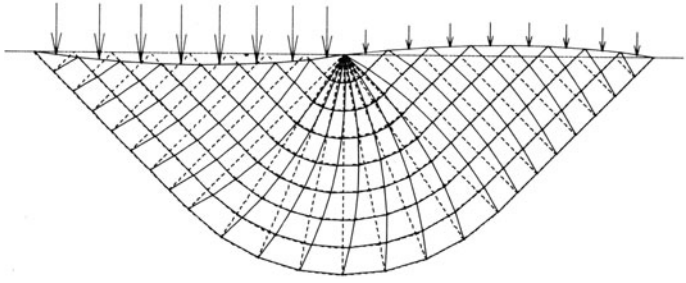
Some Typical Results Plotted.

The following plots were output by the program on a graph plotting device attached to the computer, and

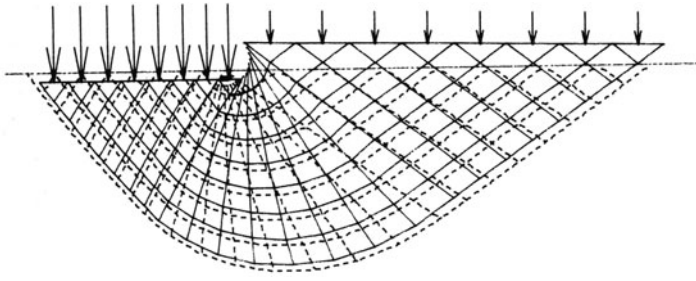
illustrate cases of undrained and drained failure.



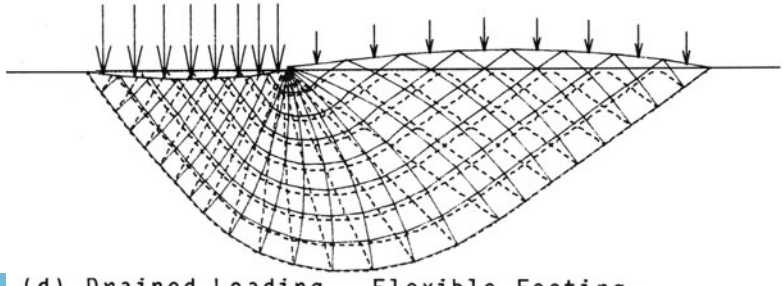
(a) Undrained Loading - Tilting.



(b) Undrained Loading - Flexible Footing.



(c) Drained Loading - Rigid Footing.



(d) Drained Loading - Flexible Footing.

Figure 9. Sample Plotted Outputs.

DISCUSSION

The associated fields method has the merit of requiring a relatively small quantity of input data, and the results obtained for bearing capacity compare favourably with those calculated by other methods. The amount of computer storage required to obtain reasonably accurate results, is also sufficiently limited to allow use of the method on modest microcomputer equipment. The program itself is also relatively compact when compared with those employing, for example, the Finite Element method, which is the main computing tool for engineering problems.

REFERENCES

Coulomb, C.A. (1773) Sur une application des regles de maximis et minimis a quelques problemes de statique relatifs a l'architecture. Acad. R. Sci. Mem. Math. Phys. par divers savants 7:343-382.
Rankine, W.J.M. (1857) On the stability of loose earth. Philos. Trans. R. Soc. London 147:9.
St-Venant, B.de. (1870) Comptes Rendus Acad. Sci. Paris, 70.
Levy, M. (1870) Comptes Rendus Acad. Sci. Paris, 70.
Massau, J. (1899) Memoire sur l'integration des equations aux derivee partielles. Reprinted as Edition du Centenaire, Delporte, Mons, 1952.
Kotter, F. (1903) Die Bestimmung des Druckes an gekrummter Gleitflachen, eine Aufgabe aus der Lehre vom Erddruck. Monatsber. Akad. Wiss. Berlin 1903:229-233.
Prandtl, L. (1921) Uber die Eindringungsfestigkeit (Harte) plastischer Baustoffe und die Festigkeit von Schneiden. Z. Angew. Math. Mech., 1(1):15-20.
Fellenius, W. (1926) Mechanics of Soils. Statika Gruntov, Gosstrollzdat. 1933.
Terzaghi, K. and Peck, R.B. (1967) Soil Mechanics in Engineering Practice. Wiley, New York.
Sokolovskii, V.V. (1965) Statics of Granular Media. Pergamon Press, New York.
Hill, R. (1950) The Mathematical Theory of Plasticity Clarendon Press, Oxford.
Schofield, A.N. and Wroth, C.P. (1968) Critical State Soil Mechanics. McGraw-Hill, New York.

ACKNOWLEDGEMENTS

Thanks are due to Dr. G.T.Houlsby of the Engineering Science Dept. at Oxford University, for valuable suggestions in connection with velocity field boundary conditions, and to our Heads of Departments for allowing the opportunity for research work.

MICROCOMPUTER ANALYSIS OF PREFABRICATED TUBULAR SCAFFOLDING

by

Grant T. Halvorsen
Department of Civil Engineering
West Virginia University
Morgantown, WV 26506 USA

and

Thomas J. Parsons
Division of Safety Research
National Institute for Occupational
Safety and Health,
Morgantown, WV 26505 USA

INTRODUCTION

In the United States a number of dramatic construction failures have occurred during the past decade. These failures have drawn attention to many problems dealing with construction safety. Although the catastropic failures have garnered much attention, there are many less dramatic failures which make the construction industry second only to mining in terms of worker risk of death or serious injury. Federal agencies including the National Institute for Occupational Safety and Health (NIOSH), and the National Bureau of Standards (NBS) are actively involved in research focused on improving safety during construction. One research area at NIOSH has been the application of microcomputers to construction safety problems.

With regard to scaffold systems, the U.S. Occupational Safety and Health Administration, OSHA, (1981) estimated that approximately 32000 scaffold-related injuries occur annually. The Bureau of Labor Statistics conducted a survey of scaffolding accidents in 1978. The results indicate that about 30 percent of scaffold accidents occur with prefabricated tubular scaffolds, and of these, approximately 30 percent are related to structural failures of cross bracing, work platforms, or supports.

In North America, prefabricated tubular scaffold systems are used extensively in the concrete and masonry trades for both new construction and maintenance. This system is based on welded tubular frames which are produced under factory conditions, and may be reused indefinitely. At a particular job site the frames are assembled vertically, then braced laterally by a combination of horizontal or diagonal bracing elements, or by ties into the construction. Figure 1 illustrates two typical scaffold frames, the step-type frame, which features a built-in ladder, and the open-type frame.

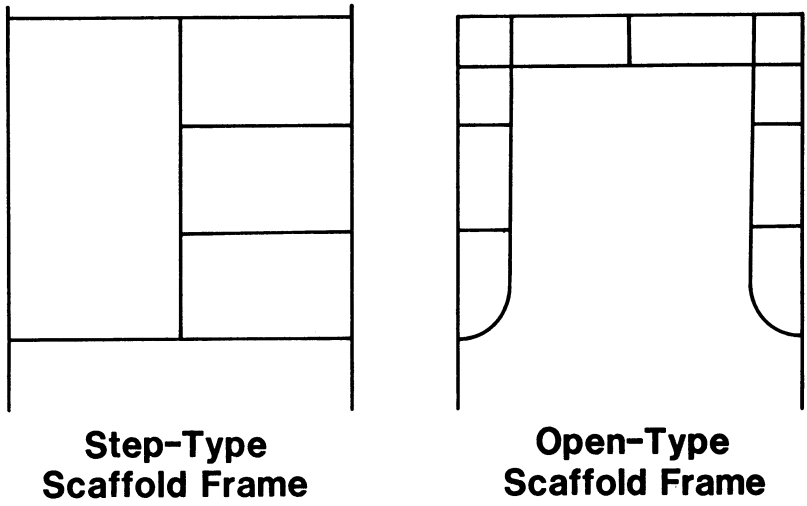


Figure 1 -- Typical Scaffold Frames

The systems are normally used in a pre-engineered or non-engineered basis. Applicable safety codes do not presently require that scaffolding be designed, or its erection supervised, by a licensed professional engineer unless the scaffold height approaches 40 metres. This is true of standards developed through the consensus process, such as American National Standards Institute Standard ANSI A10.8, "Safety Requirements for Scaffolding," and government regulations, such as the U.S. Occupational Safety and Health Administration (OSHA) regulations. The lack of regular involvement by a design professional has resulted in the development of informal guidelines, and word-of-mouth training with respect to scaffold erection procedures. Consequently, scaffold systems with questionable integrity are routinely found on construction sites. Some recent, limited - scope field studies (Hunt, 1980) indicate that typical loads applied to prefabricated tubular scaffolding systems may greatly exceed those anticipated in the standard codes of practice. There is, accordingly, a significant

concern for the actual safety margins which exist in typical scaffold systems. Most standards applicable to prefabricated tubular scaffolds require that the scaffold safely support four times the maximum intended load.

The safety margin in scaffold systems may be influenced significantly through the absence of bracing elements. Codes of practice normally give only general guidance about bracing, such as the Scaffold Industry Association (1975) statement, "Bracing: Each frame or panel shall be braced by horizontal bracing, cross bracing, or any combination thereof for securing vertical members together laterally." Similar language is found in ANSI A10.8. Although such performance statements are commendable, they are certainly open to considerable interpretation by competent structural engineers, not to mention trades people who have little or no appreciation for the need for bracing.

SCOPE

This paper reports research conducted in progress on the behavior of prefabricated tubular scaffold systems. The goals of this research are two-fold. First, the inherent safety margins for typical scaffolding assemblies will be assessed. Second, analytical methods will be studied with respect to potential implementation on microcomputer systems for use at the construction site by a project engineer or engineering technologist.

Safety codes such as ANSI A10.8 require that scaffold systems be secured to the building or structure at intervals not exceeding 9 metres horizontally and 8 metres vertically. For typical scaffold components, the largest system that would not be tied in would consist of four to five panels vertically and four to five bays horizontally. The proposed analysis should handle problems of this magnitude, or larger.

The analyses considered in this paper are generally illustrated with step-type scaffold frames. The analytical tools are general, and may be applied just as well to open-type scaffold frames. However, from a standpoint of behavior, the step-type scaffolds are more interesting, in that any ledger load (a load applied on the top horizontal members of the frame) will induce a sidesway response in the frame. This response is due to the nonsymmetrical geometry of the step-type scaffold frame.

MICROCOMPUTER SYSTEM

The work described in this paper was conducted using the Apple II+ microcomputer. This computer has 48 K bytes of random-access-memory (RAM), extended to 64 K RAM by an Apple Language System RAM card. The net usable memory is approximately 56 K. In addition to the onboard memory, the Apple Disk Operating System 3.3 was used with 125 mm floppy disks, or a Corvus 5.7 megabyte hard disk. The hard disk is

equivalent to 37 single-sided diskettes, any of which may be accessed at a given time. With respect to operation of the computer programs developed in this research, the floppy disk and hard disk systems are functionally identical. However, use of the hard disk system permits more rapid execution, since the physical disk read and write operations are accomplished more rapidly.

TWO-DIMENSIONAL ANALYSIS

Direct stiffness analysis of 2-D frames

At the outset of this research, a decision was made to first investigate the in-plane behavior of scaffold frames, then consider three-dimensional analysis. This approach was chosen for several reasons. The major reason is that the cross bracing has two quite separate functions. One is to resist lateral loads perpendicular to the scaffold frames; a 3-D analysis must be considered to assess these forces. A second function is to provide lateral restraint for the compression elements in the scaffold frame. This function may be considered in the interpretation of the structural analyses, so a two-dimensional solution can still be quite useful. Another reason to approach the problem in stages is to provide information with respect to the ability of the Apple II+ to handle structural analysis problems by stiffness methods.

The first analysis program developed in the study was a 2-D frame analysis program based on the direct stiffness method, and solution of the equations with a simple Gaussian equation solver.

To describe a single scaffold panel as a 2-D frame structure, a total of 12 joints and 15 beam elements are required, as shown in Figure 2. An assembly of these frames four tiers high would require 42 joints and 60 elements. The simple frame analysis program was found to be capable of handling structures with 20 to 25 joints before available RAM storage was exhausted. At this point the equation solver and means of storing the global stiffness matrix are a matter of concern, and alternate means may be considered. One possibility is to store only the coefficients within the half-band width of the diagonal in the global stiffness matrix, and solve for the unknown displacements by the Choleski method. Beaufait, et al. (1975) estimate the stiffness matrix half-band width for plane frame elements as:

HBW = ((DJN + 1) * 3 - 1) (Equation 1)

where HBW is the approximate half-band width and DJN is the maximum difference in joint numbers in the structure. For the frame model in Figure 2, this leads to a half-band width of 23. For the available array storage in RAM, this

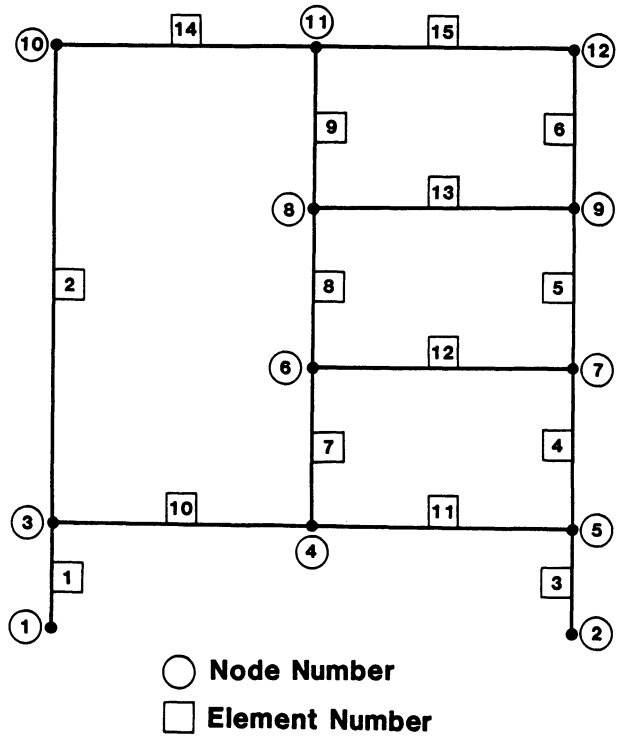


Figure 2 -- Frame Model of Scaffold Element

would permit the solution of planar assemblies of perhaps six scaffold frames (60 to 65 joints), which would satisfy one goal for the analytical methods.

However, the regularity of the scaffold frames suggests the possibility that a single structural element might be defined to describe the behavior of an entire frame. Such a "superelement" is illustrated in Figure 3. The 2-D superelement has four nodes, considering only those nodes where connections can be made to other elements, and a total of 12 degrees of freedom. A similar approach was taken by Hunt (1980), who determined scaffold frame and cross brace forces using the general-purpose finite element program FINITE, which allows user-defined substructures.

The utility of the superelement is shown in Figure 4, where superelements with 10 nodes are shown to be equivalent to 42 nodes and 60 elements.

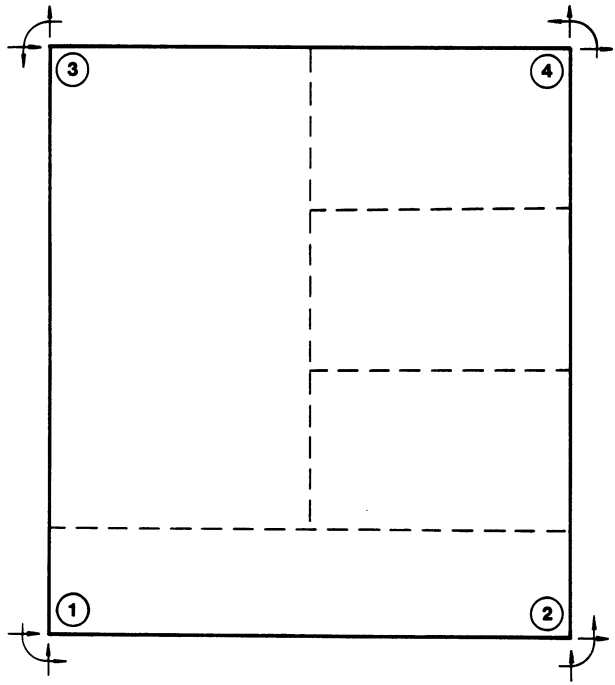


Figure 3 -- Superelement

Based on the experience with 2-D frame analysis, the use of superelements would allow full matrix storage of the stiffness matrix for perhaps 5 or 6 scaffold frames, or half-band storage of as many as 15 frames.

Development of the superelement

The development of the 12 degree-of-freedom superelement can be approached in straightforward fashion from the idea of static condensation, as described by Przemieniecki (1968). The force-displacement equations for the entire structure may be stated in global coordinates as

$$\{P\} = [K]\ \{U\} \qquad \text{(Equation 2)}$$

where $\{P\}$ is the vector of applied joint loads (including equivalent nodal forces), $[K]$ is the global stiffness matrix, and $\{U\}$ is the vector of unknown joint displacements.

If some of the applied joint loads are known to be zero, it is convenient to consider the partion of the equations

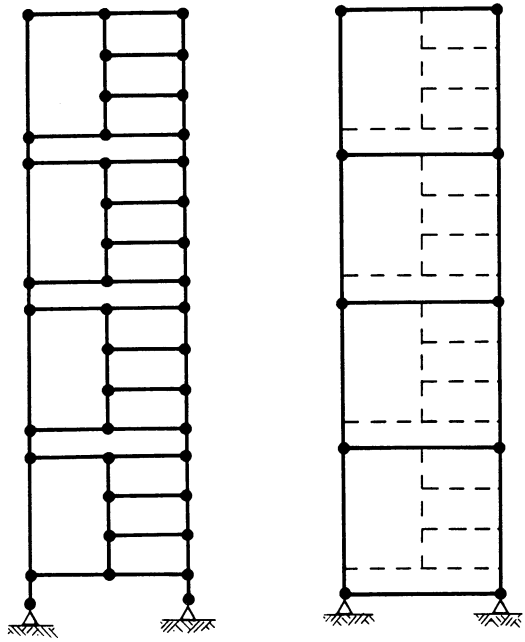


Figure 4 -- Model by Frame Element and Superelement

$$\begin{Bmatrix} P_x \\ P_y \end{Bmatrix} = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix} \begin{Bmatrix} U_x \\ U_y \end{Bmatrix} \qquad \text{(Equation 3)}$$

where $P_y = 0$, and $P_x <> 0$.

From this first condition it follows that

$K_{yx}\ U_x + K_{yy}\ U_y = 0$

Hence, $U_y = - K_{yy}^{-1}\ K_{yx}\ U_x$, if K_{yy} is non-singular.

Substituting this relationship into the partitioned equations,

$P_x = (K_{xx} - K_{xy}\ K_{yy}^{-1}\ K_{yx})\ U_x$

$= K_c\ U_x;$

where the condensed structural stiffness matrix

$$K_c = K_{xx} - K_{xy} \; K_{yy}^{-1} \; K_{yx} \qquad \text{(Equation 4)}$$

Implementation of the analysis

In the current implementation of this analysis scheme on the Apple II+, two separate programs are used to conduct the analysis. The first program, STIFFNESS.GENERATOR, is used to describe the properties of a particular scaffold frame super-element, and store the element properties on disk in an element library. An engineer could easily develop the element library for all available scaffold types well ahead of the construction phase of a particular project, or a manufacturer might develop the element information as a service to its customers. The second program, SUPER.SCAFFOLD, is like other structural analysis programs based on the direct stiffness method. Joint information can either be entered manually, or generated by the program. The user then specifies the type and location of elements used in the assembly of scaffold panels. For example, an open-type panel might be used at grade level, and step-type frames used above. From this point the program retrieves the condensed element stiffness matrix from disk storage, assembles the stiffness matrix, and solves for the unknown displacements. Backsubstitution is then performed to obtain member axial forces and end moments. Although not attempted at this point, a checking routine could be incorporated to flag overstressed members.

RESULTS

At the publication deadline for this paper, the scaffold analysis program SUPER.SCAFFOLD is not fully operational; hence no examples are provided at this time. A supplement to this paper will be made available by the authors.

EXTENSION TO THREE DIMENSIONS

In order to evaluate various cross bracing systems, a 3-D analysis is required. The concepts of the previous sections may be employed, except that 3-D frame elements are used, rather than 2-D elements. The 3-D superelement for a scaffold frame is illustrated in Figure 5.

Generally, a 3-D frame analysis considers 3 translational and 3 rotational degrees of freedom at any joint. Thus some 96 degrees of freedom are required for the stiffness matrix of the left structure in Figure 5, while the superelement would require 48 degrees of freedom. This might be reduced to 28 by considering that practical connection details permit no torsional moment transfer at the connections between frames, no moment transfer of any kind at the cross bracing points, and that the cross bracing can accept no force in the plane of the scaffold frame. Even with the half-band storage of the stiffness matrix, and the Choleski solution procedure, it is unlikely that problems of the desired nature can be

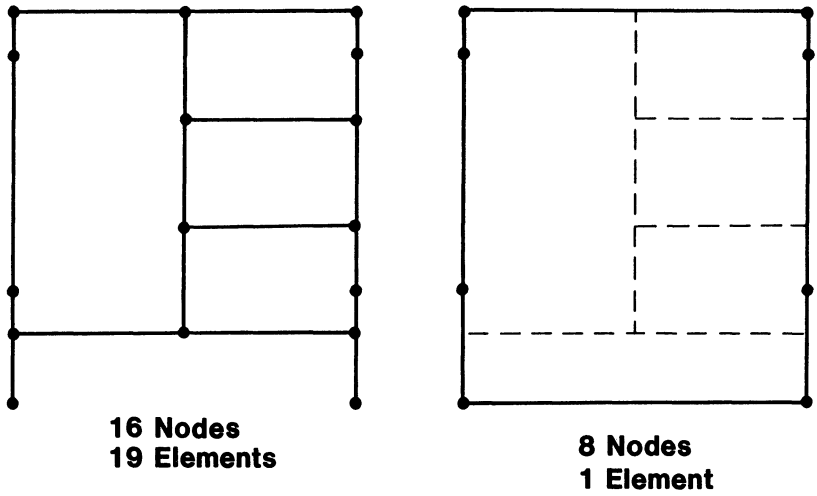


Figure 5 -- Models for 3-D Frame Analysis

solved.

SUMMARY AND CONCLUSIONS

Scaffolds are important temporary structures which have little or no involvement by the design professional. Typically, a structural analysis of prefabricated tubular scaffolds is a large, complex problem. Many joints are necessary to fully define the topology of the structure, resulting in a large number of unknown displacement coefficients, and a large stiffness matrix which must be maintained in storage. If the problem is approached by the direct stiffness method, storage of the global stiffness matrix is a critical factor in determining the problem size which may be handled by any computer. The problem is acute for the microcomputer which might conceivably be used to perform on-site structural analysis of scaffolds. It is essential that efficient storage and solution algorithms be employed to increase system capacity. For regular, repetitious structures such as scaffolds, it is convenient to describe the structure in terms of panels, or superelements. By this approach a single scaffold panel can be described by only 4 nodes and 12 degrees of freedom, in the case of 2-D analysis, as compared to 12 nodes and 36 degrees of freedom required for individual beam elements. A larger problem size can be accommodated, although the programming becomes somewhat more complicated, and an element library for available scaffold

frames must be developed. With this approach significant problems can be solved within the constraints of 48 to 64 K RAM.

Extension to 3-D problems poses some difficulties for microcomputers such as the Apple II+ where RAM is limited to 64 K, if all of the equation solution is to be done at one time. However, the availability of other microcomputers with 256 to 512 K RAM indicates that the methodology may have some very useful applications in the construction field, with the eventual aim of improving safety on the job site. Another possibility is to formulate a solution procedure where only a segment of the stiffness equations are considered in main storage at any one time. Although this would prove quite cumbersome when using floppy disks, it might be an attractive possibility where a hard disk or disk emulator were employed.

The research effort described herein is a continuing effort which includes the development of microcomputer programs, structural analysis of scaffold systems with general purpose analysis packages, and the development of simplified, approximate analysis to determine forces in scaffold frames and bracing elements.

ACKNOWLEDGMENT

Any opinions reported herein are solely those of the authors, and do not necessarily reflect those of any United States Government agency. Use of manufacturer's names does not imply the endorsement of a particular product.

This research was initiated while the first author was on a summer appointment as Civil Engineer, Division of Safety Research, NIOSH. It has been continued as cooperative research between NIOSH and West Virginia University which includes the MSCE problem reports of W.S. Easterling, III, and R. Abaszadeh. The support of J.A. Oppold, Director, Division of Safety Research, and R.C. Jensen, Chief, Accident and Injury Epidemiology Branch are gratefully acknowledged.

REFERENCES

American National Standards Institute, "American National Standard A10.8 - 1977, Safety Requirements for Scaffolding," 1977.

Beaufait, F.W., et al., Computer Methods of Structural Analysis, Department of Civil Engineering, Vanderbilt University, Nashville, Tennessee.

Bureau of Labor Statistics, "Worker Injury Reports for the Period May - November, 1978", U.S. Department of Labor, Washington, D.C.

Fattal, S.G., et al., "Review of Current Codes and Standards for Scaffolds," National Bureau of Standards Report No. NBSIR 79 - 1937, April 1980.

Hunt, B.J.P., "Development of a Research Program for Scaffolding Standards," Doctor of Engineering Dissertation, University of Kansas, 1980.

Lew, H.S., et al., "Review of Technical Information on Scaffolds," National Bureau of Standards, Report No. NBSIR 81 - 2265, May, 1981.

Occupational Safety and Health Administration, Federal Register, Volume 46, No. 11, Friday January 16, 1981.

Przemieniecki, J.S., Theory of Matrix Structural Analysis, McGraw-Hill Book Company, New York, 1968.

Scaffold Industry Association, "Code of Safe Practices for Erecting & Dismantling of Steel Scaffolds," Van Nuys, California, 1975.

STRUCTURAL ENGINEERING - MICROCOMPUTER APPLICATIONS

PRACTICAL APPLICATION OF MICRO COMPUTERS IN A STRUCTURAL CONSULTANCY

J. Seifert and Dr. M. J. Wyatt
Associated British Consultants (Computers) Ltd

Synopsis

The paper gives a case history of the experience gained by a typical firm of Consulting Engineers in using computers in the work of the Practice.

Perhaps not so typically all of the program development has been done in-house and the experience gained over the years has proven to be invaluable in assessing how the new generation of micro computers can be used to the greatest effect.

Introduction

In the mid-1960's the authors' firm started the development of an integrated system covering the analysis, design and schedule detailing of reinforced concrete slab and beam structures (Refs. 1,2,3).

The impetus for the projection was twofold, partly from a realization of the potential value and importance of computers to a consultancy practice and partly from a need to improve productivity. An analysis of the practice workload showed that much of the time was being spent on the production of reinforced concrete drawings and bending schedules. This was essentially a repetitious task carried out mainly by the young graduates and the more experienced technician detailers. The latter were for various reasons becoming a scarce commodity in the market place, which in turn made the task of training the graduates more difficult.

The equipment available at the time dictated that the system be written for a large main frame

Introduction - Continued:

computer and used on a batch 'over the counter' bureau basis. Similarly the existing technology pointed to the use of a line printer for all output. Thus a computer printed schedule defined bending and fixing instructions and this was read in conjunction with a set of A4 size standard drawings and details.

Once initial consumer resistance was overcome the system proved to be successful in both the design office and on site. Since 1968 some 200 jobs, involving many thousands of tons of reinforcement have been completed using the computer detailing system.

However, despite this success, the system had a major disadvantage in that the Engineer was kept remote from the computer which remained a "black box". Additionally reinforced concrete only represented part of the total practice workload, and few other primary tasks were identifiable as being suitable for solution by the main frame 'sledge hammer' batch solution.

It became obvious that the job engineers needed to have direct access to the computer system both in order to gain the full benefit from its use, and to help overcome the computer barrier that existed then, and to some extent still persists today.

The consultancy practice consists of a number of branch offices in different areas of the United Kingdom and overseas; thus any practical solution needed to provide a means whereby the majority of the offices could use the new technology. The management's prime consideration was for overall economy and cost effectiveness and a number of alternatives, including terminals, were examined and rejected.

Finally in 1978 the low cost micro made its appearance in the United Kingdom with machines such as the Commodore Pet, Apple 1 and the Tandy TRS 80 Model 1. This meant that a system was available which potentially could satisfy many of the identified needs and a 16K TRS 80 with cassette drive was purchased to assess the feasibility of using such a system within the design office.

Initial Development

The first experimental program was written to design simply supported steel beams and stanchions to BS 449 and had built into it the tabulated geometric properties of the main steelwork sections. The answers were displayed on the screen and hand written onto preprinted calculation sheets.

The program input was based on providing the answers to a series of questions which were displayed one after the other. This technique was employed in many contemporary programs and was based on the limitations of a teleprinter terminal. The inherent disadvantage is that once a mistake is made, it is necessary to restart the process from the beginning. As stated, it was at that time a common approach, and can be classified as the 'Teletype Syndrome' - it is indeed an approach that still persists today under the title 'Interactive Design'!

The steelwork program was developed for a particular contract and saved considerable time and effort and it also showed that it was quite feasible to use a micro computer in the office. While this is self-evident now, it was very much a stab in the dark in 1978. Such is progress.

Concurrent with this work a program was written to record the names and addresses and telephone numbers of all staff, the output being a staff list with people listed office by office. A trivial task but with 14 separate units a necessary one. It meant that, for the first time, it became practical to issue at regular intervals, an up-to-date staff list neatly typed to a consistent standard.

Following the initial success a further micro was obtained for use by a branch office, and one of the results of this was that a truss analysis program was developed by a sandwich student, based on work he was doing at University. Whilst the program had a number of serious shortcomings it worked in principle and was taken over by the central computing unit for further development. The truss program suffered from the same input problems as the steel-work program mentioned previously, in that any error meant it was necessary to return to the start of the program.

Development Stage 2

During this time work had started on a schedule detailing program for flat slabs and a drawing issue control program required for an overseas contract with over 13,000 drawings. The drawings, produced in 5 separate locations, could be issued to any one of 14 different locations with up to 10 different revisions.

Experience gained on these projects led to a new data input technique using preprinted data sheets which were then simulated on the VDU screen.

Routines were written which allowed cursor control of items of data so that any individual field could be corrected and lines of data could be added or deleted. The technique also allowed facilities to be incorporated which catered for automatic repetition of data. The data could also be recalled for later amendment as required.

Subsequently this has lead to the development of a general set of routines for screen input. These routines have much in common with modern word processing techniques. This has not only meant that programs can be developed more quickly, but also that the initial problems caused by data correction have been eliminated, with a resultant saving in terms of design time, cost, and last but not least, engineers' frustration.

In line with this approach, output was standardized to a vertical A4 format with each sheet being appropriately headed and each column being carefully titled with its appropriate units. The volume of output for each program and the degree of accuracy required for each result was kept to the minimum consistent with the designers real requirements.

Having developed a uniform input system, the next stage was to produce programs for the analysis of surface water drainage systems, simple plastic portal frames and continuous beam analysis to CP 114.

It was interesting to note that the CP 114 analysis program was rewritten in Basic from the Fortran mainframe program that was part of the original R.C. suite. The initial rewriting was done over a weekend and the program was tested and running a week later which indicates the power of the micro in software development.

Development Stage 2 - Continued:

In order to speed up the programs the use of a Basic Compiler was investigated and resulted in a considerable saving in program run time. For example the run time for the analysis of a 50 member truss with all forces and deflections was reduced from 20 minutes to 1 minute.

At this stage, with a number of working programs available, further model 1 machines were purchased for the branch offices.

A more powerful TRS 80 model II with greater disk capacity was obtained to help with the administration. Administration software currently includes:-

1. A job costing suite
2. Staff personnel program
3. Fee reminder program
4. Job list program
5. Client file program
6. Speculative work analysis program.

The fee reminder program is in the process of being extended to cover the production of Bills and Fee Ledger Balances to form a further integrated suite.

The ABCONS System

It was appreciated when CP 110 was originally announced that a replacement would be required for the CP 114 R.C. analysis, design and detailing suite. A main frame computer version was considered and indeed some initial work carried out, but it was rejected partly on the grounds of cost and partly on philosophical grounds, in that it was considered the wrong approach for the future.

It was therefore agreed that a stripped down version of the complete system would be written for the micro, this involved two main changes in the way the system would be used. Firstly instead of detailing whole floors of buildings the micro version would consider the floor as a number of slab and beam strings. This meant that the slab reactions had to be input for the beams rather than being automatically transferred within the program. However it was considered that the increased flexibility and availability of the micro would balance the extra effort required. Secondly the original system was completely integrated with analysis, design and detailing following sequentially in a continuous process whereas the micro system will allow a degree of interaction at the interface stages.

The ABCONS System - Continued:

It is significant that only four years after the introduction of the TRS 80 model I the latest Tandy 16 bit machine coupled to a hard disc will provide capacity to revert to an integrated system.

The conceptual thinking behind the original system was however carried through to the micro version. The design engineer conceives the structure in terms of the overall geometry of column layout, framed by a suitable arrangement of slabs and beams. The initial design is developed using a combination of experience, rough sketches and back of the envelope calculations to establish the member sizes.

The information from this initial design stage can then be abstracted and entered onto input data forms.

Each run can consist of up to 100 members and 50 data sheets, with each beam or slab string containing up to 10 members. In order to ensure that data is correctly entered an extensive data check program processes the data with checks for typing accuracy against a manual check as well as field range validity, and checks against various CP 110 requirements.

A comprehensive file maintenance system has been employed to enable a check to be kept on the various files. This employs a system directory controlling a hierarchical structure of general and member files, enabling the user to keep a control of the data and allowing all output to be cross referenced.

It is eventually intended to produce an index which will cross reference each element to its appropriate analysis or design, or detailing output sheet.

The system is being developed in three phases:-

1. Analysis
2. Design
3. Detailing.

The analysis phase provides a complete moment envelope covering all the pattern loadings given in CP 110 and can be either an ultimate or service load analysis. A sophisticated routine allows redistribution of up to 30% to be included if required, again in accordance with the code requirements.

The ABCONS System - Continued:

Beams can be analysed either as partial sub-frames or continuous over pin supports. All types of normally occurring loads can be included.

The output includes bending and shear force envelopes, support reaction and moment envelopes, critical design moments and shears and their position along the element.

The design phase output includes all the necessary information for a detailer to produce the appropriate drawings and includes the bar patterns and diameters, stopping off points for main tensile bars and the groupings, diameters and centres for the links. It covers all the multitudinous requirements of CP 110 in terms of bond, spacing, cut off lengths and shears.

The detailing section was at the system design stage at the time this paper was prepared. Initially it will follow the procedure used in the original system and will be based on scheduled output to be read in conjunction with standard details. One difference from the CP 114 version will be the use of the 'loose' bar system of detailing which it is considered provides greater flexibility in fixing reinforcement, simplifies the beam/column junction detail and allows a greater degree of cage pre-fabrication.

The Micro in the Design Office

When it was decided to provide certain of the design offices with computers, the decision had to be made as to which offices should be chosen initially. Two criteria were adopted; enthusiasm for having a computer from the engineering manager running the office, and some degree of computer expertise within the office. Late arrivals in the enthusiasm stakes have been told they will need to wait until the next round of machine purchases.

The application of these criteria mean that currently five offices have a in-house computer facility. There is a central computing unit which provides a back-up service, contains professional computing experience and buys all the equipment. It is responsibile for all maintenance and for the bulk buying, where appropriate, of computer sundries such as disks and paper.

The Micro in the Design Office - Continued:

It has been found essential to have one person in each office responsible to the central unit, and all communciations in the way of updated or new disks, newsletters, manuals etc., are channeled through this representative. Similarly he is expected to provide feedback on the suitability and useability of the programs and also point out the occasional bug.

Close liaison is maintained with the offices in two ways, by regular visits to the office and by regular meetings attended by the branch representatives and the central unit. The meetings are also attended by other people within the organization who have a particular interest in computer activities.

Two separate user groups have been established, one is concerned with technical applications and the other with administrative functions. The requirements for these are sufficiently different that it is considered better that "the twain shouldn't meet".

Particular individuals in certain offices have been found to have a special interest and bent towards computing. This has been encouraged and whereas an individual is free to spend his evenings as he wants, the people concerned are encouraged to discuss their ideas and the applicability to the work of the practice.

A number of the ideas have been developed to a point where it has been possible for the central unit to take them and turn them into useful working documented programs. It has to be emphasised that there is a very considerable gap between these two stages.

One further aspect that should be mentioned is the use of games as a means of obtaining acceptability of the computer. Whereas sitting down at the machine with a users manual proves daunting to many people, sitting at the same machine to play a game, even a complicated one, is acceptable and enjoyable to almost everyone.

The Trainee Engineer

Obviously all engineers, be they young trainees or chartered engineers, are encouraged to make use of the computer facility that is available. For the young engineer in particular, the presence of the computer allows him to try various alternative solutions and test the sensitivity of the structures to various types of loading and stiffness arrangements. From an examination of the results he can obtain a "feel" for the right answer.

At least that is the theory and is an oft quoted advantage for using computers, in practice, it may not work out quite so satisfactorily. There is a growing tendancy to assume that because the computer can work to six decimals and produce reams of neatly typed calculations, the results are both right and accurate. The sophistication and pseudo accuracy hides the true lack of knowledge engineers have of both actual behaviour and loading. It will be an unfortunate day when the statement is made "that it must be right because the computer says it is"! It will be an even more unfortunate day if the statement is believed!

In providing training for the young engineer it is essential that proper attention is given to the traditional techniques of producing 'back of the envelope' calculations. With this back-up expertise the answers from the computer system can be kept in perspective, and ensurance can be gained that the designer has a true feel for the results.

At present the trained and experienced human mind still has the edge in spotting the unusal error.

The Future

The immediate future as far as the micro is concerned would seem to involve an increase in all aspects of power and capacity particularly disk capacity, for no appreciable increase in real price.

Each organization will need to decide whether or not it can make proper economic use of the larger machines, bearing in mind the possible costs involved in converting existing programs.

The future - Continued:

The availability of low cost A3 size plotters is likely to lead to an increase in this form of output. Although there are many available applications for plotters the traditionally high cost of the equipment takes it out of the affordable price range for many firms.

The more distant future, which may not be that far off, is likely to see the introduction of low cost interactive graphics work stations into the design office and work patterns will need to be adjusted accordingly.

The only future certainty is that computing will play an increasing part in every aspect of the work in the design office. It is upto the individual engineer to ensure that the results are truly cost effective.

References

1. L L. Kenchington : Drawing and detailing by automated procedures - Conference Birmingham University April 1970.

2. J. Seifert : Field experience in replacing conventional detailing drawings Concrete Society Symposium- Bristol University December 1970.

3. L. L. Kenchington J. Seifert : The place of Design, Optimization and Computers Reinforced Concrete Construction University of Aston - September 1973

```
ABCONS                    SYSTEM MENU                    BMSLB V2.0
ENTER YOUR SELECTION
?..

SELECT FROM THE FOLLOWING OPTIONS:-

1. ENTER GENERAL DATA (FORMS 10-30)
2. ENTER MEMBER DATA (FORMS 40-100)
3. ANALYSIS
4. DESIGN
5. SCHEDULE
6. MAINTAIN FILES
7. INITIALISE DATA DISKETTE

 * YOU MAY LOAD A DATA DISK WHILST THIS MESSAGE IS DISPLAYED *
```

FIGURE 1. SYSTEM MENU AS DISPLAYED ON THE VDU

```
ABCONS                FORM 50 (PAPER/PAPERA)                SHEET  4
                  <SHIFT> (D)ELETE LINE,(H)ELP,(I)NSERT,(R)EPEAT
MEMB.:BAR ARRGM'T:PRE:MAX:MIN:LINK LEGS:LINK:    COVERS
REF.  :L1 :L2 :L3 :DIA:DIA:DIA:DES :NOM :DIA :TOP:BTM:S1 :S2
A101  :5  :2  :   :25 :40 :12 :4   :4   :8   :30 :40 :30 :30     A
B101  :*  :*  :   :*  :*  :*  :*   :*   :*   :*  :*  :*  :*      B
C101  :*  :*  :   :*  :*  :*  :*   :*   :*   :*  :*  :*  :*      C
D101  :*  :*  :   :*  :*  :*  :*   :*   :*   :*  :*  :*  :*      D
      :   :   :   :   :   :   :    :    :    :   :   :   :       E
      :   :   :   :   :   :   :    :    :    :   :   :   :       F
      :   :   :   :   :   :   :    :    :    :   :   :   :       G
      :   :   :   :   :   :   :    :    :    :   :   :   :       H
      :   :   :   :   :   :   :    :    :    :   :   :   :       I
      :   :   :   :   :   :   :    :    :    :   :   :   :       J
0404  :5  :2  :   :25 :40 :12 :4   :4   :8   :3  :4  :3  :3      Z
```

FIGURE 2. TYPICAL BEAM DESIGN DATA SHEET AS DISPLAYED ON THE VDU

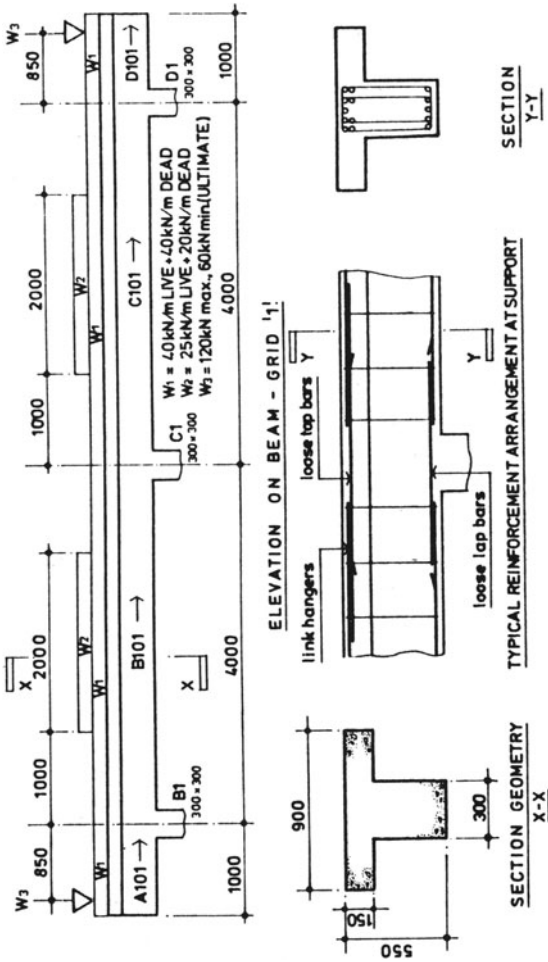


FIGURE 3. ABCONS DESIGN EXAMPLE

KENCHINGTON LITTLE & PARTNERS
BUILDING AND CIVIL ENGINEERING CONSULTANTS
CASTLE PLACE NOTTINGHAM NG1-6EN TEL:(0602)412220

JOB : SHOPPING CENTRE JOB NO. :PA/100 PART: RA/ 1
PART: 1ST FLOOR BEAMS AND SLABS PREPARED:MJW PAGE: 1/ 2
BEAMS ON GRID '1' DATE :10/11/82 REV:

REDISTRIBUTED BM ENVELOPE - UNITS: KNM

MEMBER	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
A101	0	-1	-9	-24	-40	-58	-77	-97	-118	-141	-165
	0	0	0	0	0	0	0	0	0	0	0
B101	-208	-87	-41	-20	-8	-6	-15	-34	-62	-129	-247
	0	0	52	118	154	164	152	109	37	0	0
C101	-247	-129	-62	-34	-15	-6	-8	-20	-41	-91	-208
	0	0	23	95	143	164	154	118	52	0	0
D101	-165	-141	-118	-97	-77	-58	-40	-24	-9	-1	0
	0	0	0	0	0	0	0	0	0	0	0

REDISTRIBUTED SF ENVELOPE - UNITS: KN

MEMBER	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
A101	0	-13	-145	-158	-170	-183	-195	-208	-220	-233	-246
	0	0	0	0	0	0	0	0	0	0	0
B101	0	0	0	0	0	-26	-104	-181	-245	-295	-345
	328	278	228	164	87	9	0	0	0	0	0
C101	0	0	0	0	0	0	-77	-155	-219	-269	-319
	345	295	245	181	104	26	0	0	0	0	0
D101	0	0	0	0	0	0	0	0	0	0	0
	246	233	220	208	195	183	170	158	145	13	0

SUPPORT MOMENT/REACTION ENVELOPE -UNITS: REACTIONS KN,MOMENTS KNM

SUPPORT	REACTIONS		MOMENTS				SUM BEAM
REF	MAX	MIN	+ UP	- UP	+ DN	- DN	STIFFNESS
B1	574	189	29	-44	44	-29	0.000433
C1	691	440	35	-35	35	-35	0.000867
D1	565	189	44	-29	29	-44	0.000433

SUMMARY OF RESULTS - UNITS: DIM MM,MOMENTS KNM,SHEARS KN

MEMBER	DESIGN MOMENTS			0.6-BRED			ZERO MOMENT		DESIGN SHEARS	
REF	LT	SPAN	RT	LT	SPAN	RT	LT-->	<--RT	LT	RT
A101	0	-58	-165	0.60	0.60	0.60	0	1000	0	-246
	0	0	0	0.60	0.60	0.60	0	0	0	0
B101	-208	-6	-247	0.60	0.60	0.50	0	0	0	-345
	0	164	0	0.60	0.56	0.60	400	400	328	0
C101	-247	-6	-208	0.50	0.60	0.60	0	0	0	-319
	0	164	0	0.60	0.56	0.60	400	400	345	0
D101	-165	-58	0	0.60	0.60	0.60	1000	0	0	0
	0	0	0	0.60	0.60	0.60	0	0	246	0

FIGURE 4. ABCONS ANALYSIS PRINTOUT (FOR EXAMPLE IN FIGURE 3.)

KENCHINGTON LITTLE & PARTNERS
BUILDING AND CIVIL ENGINEERING CONSULTANTS
CASTLE PLACE NOTTINGHAM NG1-6EN TEL:(0602)412220

JOB : SHOPPING CENTRE JOB NO. :PA/100 PART: RA/ 2
PART: 1ST FLOOR BEAMS AND SLABS PREPARED:MJW PAGE: 1/ 1
BEAMS ON GRID '1' DATE :10/11/82 REV:

MAIN BAR DETAILS - UNITS: DIM-MM,AREAS-SQMM

BEAM REF.	POSN/FACE	AS REQD	AS PROV	ITEM	LAYER	BARS	STARTS		ENDS	
A101	SPAN/BTM	453	565	(1)	1	2T12	RT	40	RT	820
				(2)	1	2T12	RT	40	RT	820
				(3)	1	1T12	RT	40	RT	820
B101	LEFT/BTM			(1)	1	2T16	LT	870	RT	900
				(2)	1	2T16	LT	870	RT	900
				(3)	1	1T12	LT	870	RT	550
	SPAN/BTM	859	917	(1)	1	2T16	RT	170	RT	3820
				(2)	1	2T16	RT	170	RT	3820
				(3)	1	1T12	RT	1030	RT	2910
	LEFT/TOP	1305	1457	(11)	1	2T20	LT	1440	RT	2330
				(12)	1	2T20	LT	1440	RT	2330
				(13)	1	1T16	LT	1440	RT	950
C101	LEFT/BTM			(1)	1	2T16	LT	910	RT	900
				(2)	1	2T16	LT	910	RT	900
	SPAN/BTM	859	917	(1)	1	2T16	RT	170	RT	3820
				(2)	1	2T16	RT	170	RT	3820
				(3)	1	1T12	RT	1240	RT	2960
	LEFT/TOP	1623	1923	(11)	1	2T25	LT	2510	RT	2420
				(12)	1	2T20	LT	2510	RT	2420
				(13)	1	1T20	LT	950	RT	950
BEAM REF.	POSN/FACE	AS REQD	AS PROV	ITEM	LAYER	BARS	STARTS		ENDS	
D101	LEFT/BTM			(1)	1	2T16	LT	910	RT	860
				(2)	1	2T16	LT	910	RT	860
				(3)	1	1T12	LT	550	RT	860
	SPAN/BTM	453	565	(1)	1	2T12	RT	170	RT	960
				(2)	1	2T12	RT	170	RT	960
				(3)	1	1T12	RT	170	RT	960
	LEFT/TOP	918	1457	(11)	1	2T20	LT	2430	RT	1430
				(12)	1	2T20	LT	2430	RT	1430
				(13)	1	1T16	LT	950	RT	1430

LINK DETAILS - UNITS: DIM-MM,STRESSES-N/SQMM

BEAM REF	ZONE (FROM	LH	END)	SHEAR (KN)	AS %	VC	V	ASV/SV REQD	LEGS @ CRS
A101	50	-	800	226	0.98	0.69	1.51	0.36	4T 8 @ 300
B101	200	-	970	309	0.53	0.56	2.12	1.27	4T 8 @ 150
	970	-	3010	208	0.53	0.56	1.43	0.36	4T 8 @ 300
	3010	-	3800	326	0.53	0.56	2.24	1.36	4T 8 @ 125
C101	200	-	1080	326	0.53	0.56	2.24	1.36	4T 8 @ 125
	1080	-	3180	208	0.53	0.56	1.43	0.36	4T 8 @ 300
	3180	-	3800	300	0.53	0.56	2.06	1.21	4T 8 @ 150

Figure 5. ABCONS DESIGN OUTPUT (FOR EXAMPLE IN FIGURE 3.)

CONCISE EQUATIONS AND PROGRAM FOR EXACT EIGENSOLUTIONS OF PLANE FRAMES INCLUDING MEMBER SHEAR

W P Howson, J R Banerjee and F W Williams

Department of Civil Engineering and Building Technology,
University of Wales Institute of Science and Technology,
King Edward VII Avenue, Cardiff CF1 3NU, UK.

SUMMARY

A unified notation is adopted to present concise member equations for a Timoshenko beam-column and a static beam-column with finite shear stiffness. The Fortran coding necessary to incorporate these equations into a previously published computer program is given. The resulting program enables any number of exact natural frequencies or elastic critical buckling loads of rigidly jointed plane frames to be calculated using an algorithm which ensures that none can be missed. The corresponding mode shapes can then be retrieved by a simple technique which offers improved accuracy over more established methods. The program contains approximately 250 Fortran statements and has been designed to minimise core store requirements at the expense of computation time. The use of the program as a 'black box' is fully explained with illustrative examples.

INTRODUCTION

In an earlier paper Howson (1979) presented two Fortran computer programs which calculate the eigenvalues and eigenvectors of rigidly jointed plane frames. The eigenvalues are natural frequencies and critical load factors in vibration and buckling problems, respectively, while the eigenvectors are the corresponding mode shapes. The theory was based on the stiffness method of analysis and uses classical (exact) Bernoulli-Euler member equations together with an algorithm which ensures that the required eigenvalues are found with certainty. The corresponding eigenvectors are then retrieved by a simple technique which offers improved accuracy over more established methods. Each program, which was specifically written for minicomputer and microcomputer application, contains approximately two hundred Fortran statements and their use as

'black boxes' was fully explained with illustrative examples.

More recently Lightfoot (1980) presented the exact member equations for a range of straight line elements in terms of a unified notation based on that of Blaskowiak and Kaczkowski (1966). This notation is shown in equation (1) and Figure 1.

$$\begin{bmatrix} F_1\ell \\ F_2 \\ F_3\ell \\ F_4 \end{bmatrix} = \frac{EI}{\ell} \begin{bmatrix} \gamma & \nu & -\varepsilon & \delta \\ \nu & \alpha & -\delta & \beta \\ -\varepsilon & -\delta & \gamma & -\nu \\ \delta & \beta & -\nu & \alpha \end{bmatrix} \begin{bmatrix} D_1/\ell \\ D_2 \\ D_3/\ell \\ D_4 \end{bmatrix} \qquad (1)$$

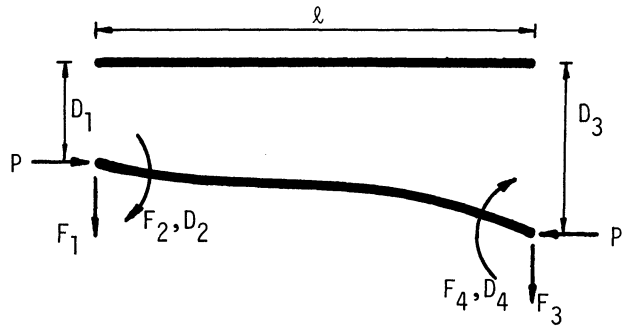


Figure 1. Member end forces and displacements.

P, EI and ℓ are the axial force, flexural rigidity and length of the member, F_1- F_4 are the transverse forces and moments at its ends and D_1 - D_4 are the corresponding displacements. The Greek symbols are the appropriate stiffness functions, e.g. stability functions in a buckling problem.

This paper extends the range of elements which Lightfoot covered by using his notation in a particularly concise presentation of the exact member equations for a Timoshenko beam-column, and of the corresponding member equations for a static beam or beam-column with finite shear stiffness. Furthermore, expressions are given for J_m, the number of eigenvalues of a member with its ends clamped which are exceeded by a trial value of the circular frequency (vibration problems) or load factor (buckling problems). These expressions are required in the algorithm used by Howson (1979) to ensure that no eigenvalues can be missed. Lightfoot gave no such expressions for J_m, although several of his elements are intended for use in buckling or vibration problems. However, since Lightfoot's relevant elements are a sub-set of those presented

herein, the expressions for J_m given below are applicable.

Finally, the Fortran coding required to modify the earlier vibration program to incorporate the new equations is presented. Illustrative examples are then used to show how the resulting program can be used as a 'black box' to obtain exact vibration or buckling eigenvalues and their corresponding eigenvectors.

VIBRATION THEORY

Timoshenko theory was used by Howson and Williams (1973) to obtain the member equations for a vibrating beam-column with finite shear rigidity, k'AG, and finite rotatory inertia in terms of the dimensionless parameters

$$\left.\begin{aligned} b^2=\mu\ell^4\omega^2/EI \quad ; \quad p^2=P\ell^2/EI; \\ r^2=I/A\ell^2 \quad ; \quad s^2=EI/k'AG\ell^2, \end{aligned}\right\} \tag{2}$$

where p^2, r^2 and s^2 represent the effects of axial force, rotatory inertia and shear deflection; ω is the circular frequency; μ and G are the mass per unit length and modulus of rigidity of the member; A, I and k' are the area, second moment of area and shape factor of the member's cross-section; and the axial force in the member, P, is positive for compression and negative for tension. Using Lightfoot's notation, a small correction due to Akesson (1976) and some changes to improve conciseness these member equations give

$$\left.\begin{aligned} &\alpha=\Gamma(SC'-j\eta CS') \quad ; \quad \beta=\Gamma(j\eta S'-S) \quad ; \\ &\nu=Z\Gamma\{(\Phi+j\eta\Lambda)SS'-(\Lambda-\eta\Phi)(1-CC')\}/(\Lambda+\eta\Phi) \quad ; \\ &\delta=Z\Gamma(C'-C) \quad ; \quad \gamma=b^2\Gamma(CS'+\eta SC')/\Lambda\Phi \quad ; \\ &\varepsilon=b^2\Gamma(S'+\eta S)/\Lambda\Phi, \end{aligned}\right\} \tag{3}$$

where the variables on the right-hand sides are given by

$$\left.\begin{aligned} &q=1-b^2r^2s^2 \quad ; \quad t=1-s^2p^2 \quad ; \\ &\Delta=(qp^2+b^2r^2+b^2s^2)/2t \quad ; \end{aligned}\right\} \tag{4}$$

$$\left.\begin{aligned} &\Phi^2=\Delta+(\Delta^2+qb^2/t)^{1/2} \quad ; \\ &j\Lambda^2=-\Delta \\ &Z=t\Phi-b^2s^2/\Phi \quad ; \quad \eta=Z/(jt\Lambda+b^2s^2/\Lambda) \quad ; \\ &S=\sin\Phi \quad ; \quad C=\cos\Phi \quad ; \\ &S'=\sinh\Lambda \ , \ C'=\cosh\Lambda \ , \ j=1 \text{ for } q>0 \ ; \\ &S'=\sin\Lambda \ , \ C'=\cos\Lambda \ , \ j=-1 \text{ for } q\leq 0 \\ \text{and} \quad &\Gamma=(\Lambda+\eta\Phi)/\{2\eta(1-CC')+(1-j\eta^2)SS'\}. \end{aligned}\right\} \tag{5}$$

Equations (3)-(5) remain valid when any combination of the effects of axial force, rotatory inertia and shear deflection are suppressed by setting the relevant parameter (p^2, r^2 or s^2) equal to zero. However, a limiting process must be used as $b^2 \to 0$, in order to obtain the following equations for buckling.

BUCKLING THEORY

The member equations for a static beam-column with finite shear stiffness are given by

$$\left.\begin{array}{lll} \alpha=\chi(S-ZC) & ; & \beta=\chi(Z-S) \quad ; \\ \nu=\delta=\alpha+\beta & ; & \gamma=\varepsilon=j\chi Z^2 S, \end{array}\right\} \qquad (6)$$

where equation (4) still applies but with $b^2=r^2=0$, and

$$\left.\begin{array}{l} \Phi^2=2j\Delta \quad ; \quad Z=t\Phi \quad ; \\ S=\sin\Phi \quad , \quad C=\cos\Phi \quad , \quad j=1 \text{ for } P>0 \quad ; \\ S=\sinh\Phi \quad , \quad C=\cosh\Phi \quad , \quad j=-1 \text{ for } P<0 \\ \text{and} \quad \chi=p^2/\{2Z(1-C)-jZ^2S\}. \end{array}\right\} \qquad (7)$$

Equations (6) and (7) remain valid for $s^2=0$, i.e. for the Bernoulli-Euler beam, although in this case more concise expressions have been given by Williams (1981). However, as $P \to 0$ it is necessary to take equations (6) and (7) through a further limiting process, which yields

$$\left.\begin{array}{lll} \alpha=1+3/(1+12s^2) & ; & \beta=\alpha-2 \quad ; \\ \nu=\delta=\alpha+\beta & ; & \gamma=\varepsilon=2\nu \quad . \end{array}\right\} \qquad (8)$$

Equation (8) gives the usual slope-deflection equations when $s^2=0$.

FORMULAS FOR J_m

Equation (1) governs flexure of members. When members are extensible the axial displacements of their ends are related to the corresponding axial forces by Hooke's Law in the static case, and by frequency dependent stiffnesses in the vibration case. These equations retain exactly the form given by Howson (1979) and do not couple with flexure, therefore they will not be discussed in detail again. However, in developing expressions for J_m it can be seen that axial vibrations are possible for a member with clamped ends, but that buckling must involve flexure. Therefore

$$\left.\begin{aligned} J_m &= J_a + J_b \quad \text{for vibration} \\ \text{and} \quad J_m &= J_b \quad \text{for buckling,} \end{aligned}\right\} \qquad (9)$$

where J_a is the contribution to J_m made by axial vibrations, and is given by Howson (1979) in the expression beneath equation (15), and J_b is the flexural component of J_m. For the vibrating beam-column of equations (2)-(5) Howson and Williams (1973) have shown that J_b is given by

$$J_b = J_c - 1 + \tfrac{1}{2}\,sg\{\alpha\} + \tfrac{1}{2}\,sg\{\alpha - \beta^2/\alpha\}, \qquad (10)$$

where

$$\left.\begin{aligned} J_c &= J_d \quad \text{for } q>0 \ ; \\ J_c &= J_d + J_e \quad \text{for } q\leq 0 \ ; \\ J_d(J_e) &= \text{highest integer } < \Phi/\pi\,(\Lambda/\pi + 1) \end{aligned}\right\} \qquad (11)$$

and $sg\{\}$ is +1 or -1 depending on the sign of the quantity within the bracket.

The method used to obtain equations (9)-(11) can also be applied to the static member of equations (6)-(8) to give

$$\left.\begin{aligned} J_m &= 0 \quad \text{for } P\leq 0 \\ \text{and} \quad J_c &= \text{highest integer } < \Phi/\pi \quad \text{for } P>0, \end{aligned}\right\} \qquad (12)$$

where equation (10) applies if $P>0$.

Equations (9)-(11) cover the two vibrating members considered by Lightfoot (1980), by setting $p^2=r^2=s^2=0$ or $r^2=s^2=0$, respectively. Similarly equations (9), (10) and (12) cover the static member considered by Lightfoot by setting $s^2=0$.

In buckling problems, the Φ of equation (7) has $t\ (=1-s^2p^2)$ in its denominator via equation (4) and so equation (12) gives $J_c \to \infty$ as $t \to 0$. Hence if any member has $t \leq 0$, the trial load factor must have exceeded an infinite number of critical load factors of that member and thus also of its parent frame. Therefore, in the program defined below, such a load factor is taken as an upper bound on the critical load factor being sought.

PROGRAM CHANGES

The listing given in Appendix II of the paper by Howson (1979) can be altered to use the member equations given above by making the following changes. Bracketed numbers and symbols in the annotation refer to equations and symbols used in the text.

(a) Line 2 should be replaced by

```
DATA NI,NR,LSF,MS,NPR,NPS,DR,FL,X/2*999,6,3,6,7,0.5,0.0,1.0E30/
```

(b) Insert the following between lines 6 and 7.

```
READ(5,1030)VI,RI,SI,PR,SF,FC
WRITE(6,1030)VI,RI,SI,PR,SF,FC
```

] Problem data, see Table 1a.

(c) Lines 36-38 inclusive should be replaced by

```
30 RA(J+6)=SQRT(RA(J+4)*RA(J+4)+RA(J+5)*RA(J+5))
```

Member length, ℓ.

(d) Insert the following between lines 154 and 155.

```
IF(FR.LT.VI*FC)STOP 2
```

See section dealing with output.

(e) Replace line 165 with

```
FP=FR/(2.0*PI*VI+1.0-VI)
```

Convert radians to Hertz in vibration problems.

(f) Replace line 213 with

```
1030 FORMAT(1X,1P6E11.3)
```

(g) Lines 55-80 inclusive should be replaced by

```
      C=RA(J+4)/RA(J+6)
      S=RA(J+5)/RA(J+6)
      AN=RA(J+6)*FR*SQRT(RA(J+2)/RA(J+1))
      AK=RA(J)/(RA(J+6)*RA(J+6))
      B2=VI*RA(J+2)*RA(J+6)*RA(J+6)*FR*FR/AK
      R2=VI*RI*AK/RA(J+1)
      S2=SI*2.0*AK*(1.0+PR)/(RA(J+1)*SF+1.0-SI)
      IF(RA(J+3).EQ.0.0.AND.VI.EQ.0.0)GOTO 76
      T1=RA(J+3)*((VI+FR)-VI*FR)
      QQ=1.0-B2*R2*S2
      TT=1.0-S2*T1/AK
      DA=0.5*(QQ*T1/AK+B2*(R2+S2))/TT
      IF(VI.EQ.1.0.AND.TT.LE.0.0)STOP 1
      IF(TT.LE.0.0)JJ=9999
      IF(TT.LE.0.0)GOTO 250
      PH=SQRT(ABS(SQRT(DA*DA*VI+B2*QQ/TT)+(2.0-VI)*DA))
      ZZ=TT*PH-B2*S2/PH
      SP=SIN(PH)
      CP=COS(PH)
      ST=1.0
      IF(VI.GT.0.0)GOTO 73
      IF(RA(J+3))71,76,72
   71 ST=-1.0
      SP=0.5*(EXP(PH)-1.0/EXP(PH))
      CP=0.5*(EXP(PH)+1.0/EXP(PH))
   72 CH=T1/(AK*(2.0*ZZ*(1.0-CP)-ST*ZZ*ZZ*SP))
      T1=CH*(SP-ZZ*CP)*AK*RA(J+6)
      T2=CH*(ZZ-SP)*AK*RA(J+6)
      IF(ST)76,76,75
   73 IF(QQ.LE.0.0)ST=-1.0
      AL=SQRT(ST*(SQRT(DA*DA+B2*QQ/TT)-DA))
      ET=ZZ/(ST*AL*TT+B2*S2/AL)
      S1=SIN(AL)
      C1=COS(AL)
      IF(ST.LE.0.0)GOTO 74
      S1=0.5*(EXP(AL)-1.0/EXP(AL))
      C1=0.5*(EXP(AL)+1.0/EXP(AL))
   74 GA=(AL+ET*PH)/(2.0*ET*(1.0-CP*C1)+(1.0-ST*ET*ET)*SP*S1)
      TT=AK*ZZ*GA/(AL+ET*PH)
      T1=GA*(SP*C1-ST*ET*CP*S1)*AK*RA(J+6)
      T2=GA*(ST*ET*S1-SP)*AK*RA(J+6)
```

] Member orientation.

Parameter for longitudinal motion.

b^2, r^2, s^2, q, t, Δ] See equations (2) and (4).

] Branch on $1-s^2p^2 \leq 0$.

Φ, z, $\sin\Phi$, $\cos\Phi$] See equations (5) and (7).

Initialise j in equations (3),(5)-(7).

Branch on vibration problem.

Branch on member axial force.

j, $\sinh\Phi$, $\cosh\Phi$, χ, α, β] See equations (6) and (7).

Branch on member axial force.

j, Λ, η, $\sin\Lambda$, $\cos\Lambda$, $\sinh\Lambda$, $\cosh\Lambda$, Γ, α, β] See equations (3) and (5).

```
      JJ=JJ+INT(AN/PI)-((INT(1.5*ST)-1)*INT(AL/PI+1.0))/2
   75 K=0                                                          Adds J_m for current member to J.
      IF(T1.LE.0.0)K=1                                             See equations (9)-(12).
      IF(T1-T2*T2/T1.LE.0.0)K=K+1
      JJ=JJ+INT(PH/PI)-K
   76 DO 80 L=1,3                                                  Assembly loop.
      K=MAXO(L-2,0)
      SM=3+2*(K-L)
      SN=SM+FLOAT(2*K)
      A3=(SN+1.0)*T1/2.0-(SN-1.0)*T2/2.0
      IF(VI.GT.0.0)GOTO 78
      A1=SN*RA(J+1)/RA(J+6)                                        Buckling.
      IF(RA(J+3).EQ.0.0)GOTO 77                                    See equations
      A2=SN*ST*CH*ZZ*ZZ*SP*AK/RA(J+6)                              (6) and (8).
      A4=SM*SN*CH*ZZ*(1.0-CP)*AK                                                  Elements of
      GOTO 79                                                                     k_11, k_12 & k_22.
   77 A2=SN*AK*12.0/(RA(J+6)*(1.0+12.0*S2))
      A3=SN*(0.25*A2*RA(J+6)+AK)*RA(J+6)
      A4=0.5*SM*A2*RA(J+6)
      GOTO 79
   78 A1=RA(J+1)*AN*COS(AN)/(RA(J+6)*SIN(AN))                      Vibration.
      A2=B2*GA*(CP*S1+ET*SP*C1)*AK/(AL*PH*RA(J+6))                 See
      A4=SM*TT*((PH+ST*ET*AL)*SP*S1-(AL-ET*PH)*(1.0-CP*C1))        equation (3).
      IF(L.NE.2)GOTO 79
      A1=-A1/COS(AN)
      A2=-A2*(S1+ET*SP)/(CP*S1+ET*SP*C1)
      A4=ZZ*GA*AK*(C1-CP)
   79 L1=IBK+(L-1)*I+K*((MA-2)*I-I*(I-1)/2)                        Pointers to the stiffness
      L2=L1+MA-K*I                                                 triangle stored in RA.
      L3=L2+MA-K*I-1
```

GUIDE TO DATA PREPARATION AND RESULTS

Input

Data should be prepared exactly as described in Table 1 of Howson (1979), except that card group 1a should be inserted between card groups 1 and 2 and card group 6 should be altered, as shown in Table 1a below.

Card group number	No. of cards in group	Fortran variable name	Format	Item	Note number	Comment
1a	1	VI	1X,1P6E11.3			Set to 1.0 for a vibration problem, zero otherwise.
		RI			1a	Set to 1.0 if rotatory inertia is to be considered, zero otherwise.
		SI				Set to 1.0 if shear deformation is to be considered, zero otherwise.
		PR			2a	Poisson's ratio.
		SF			2a	Section shape factor. See Cowper (1966).
		FC			3a	Frequency constraint factor.
6	NM	RA*	1X,1P6E11.3	EI		Flexural rigidity.
				EA		Extensional rigidity.
				μ	1a	Mass/unit length.
				P	4a	Axial force.
				x_p		Member projection on global x' axis.
				y_p		Member projection on global y' axis.

Table 1a. Changes to input data scheme.

The notes referred to in column 6 of Table 1a above are as follows.

1a. The values of μ and RI are ignored when VI is zero.

2a. The values of PR and SF are ignored when SI is zero.

3a. A small positive number, typically 10^{-3}. The program terminates in vibration problems if the circular frequency becomes <FC. FC is ignored in buckling problems. See section on output, below.

4a. Compressive and tensile forces are denoted by positive and negative values, respectively. They are the datum values in buckling problems.

Output

The style of the output for each problem is unchanged, apart from the data check which mirrors the changes required by Table 1a.

The two checks included in the original programs are unaltered, except that the necessary size of the real array, RA, has been increased by NM locations to allow for the changes required by card group 6. See Table 1a.

Two additional checks have been incorporated into the new program and both refer only to vibration problems. In the first of these the program is terminated with the message 'STOP 1' if $s^2p^2 \geq 1$, since this represents an attempted vibration analysis of a structure which must inevitably be buckled as a consequence of the arguments given in the second paragraph below equation (12). However, the structure may also be buckled when $s^2p^2<1$. In this case the program would converge on zero as the fundamental frequency and fail uncontrollably. To avoid this the program terminates with the message 'STOP 2' when the circular frequency falls below the value FC given in the data. For typical problems it is suggested that FC=10^{-3}.

Illustrative examples

A vibration problem and a buckling problem are now solved using an ICL 2980 computer with an integer word length of 4 bytes and a real word length of 8 bytes. In both cases the frame topology and member properties are precisely those of the vibration problem considered by Howson (1979), except that the horizontal members now carry a compressive axial force of $4EI/\ell^2$, the diagonal members carry a tensile axial force of $5EI/\ell^2$, where ℓ is the length of the member concerned, and the vertical members are unloaded.

In the vibration problem it is assumed that the effects of axial force and rotatory inertia are to be included, the effect of shear deformation is to be suppressed and that the first, second, third and twentieth natural frequencies are to be calculated together with the modes of vibration corresponding to the

first and twentieth. The output is shown in Figure 2.

```
    1
 1.000E+00  1.000E+00  0.000E+00  0.000E+00  0.000E+00  1.000E-03
  4  1.0000E+01  1.0000E+06   9   1   6   5
   -1   2   3 -20
  1   0   1   1   2   3   1   3   4   2   4   4   2   5   6
  2   3   5   2   0   2   3   5   4   4   5   5
    1   1
 5.000E+06  9.000E+08  3.500E+01  0.000E+00  0.000E+00 -3.000E+00
 5.000E+06  9.000E+08  3.500E+01  1.250E+06 -4.000E+00  0.000E+00
 5.000E+06  9.000E+08  3.500E+01 -1.000E+06  4.000E+00 -3.000E+00
 5.000E+06  9.000E+08  3.500E+01  1.250E+06  4.000E+00  0.000E+00
 5.000E+06  9.000E+08  3.500E+01  0.000E+00  0.000E+00  3.000E+00
 5.000E+06  9.000E+08  3.500E+01 -1.000E+06  4.000E+00  3.000E+00
  1  2.5307E+01
-3.443E-23  5.878E-02  8.937E-02
 1.532E-01  5.248E-01  1.933E-01
-5.157E-02  5.353E-01  1.584E-01
 1.512E-01  1.000E+00  1.927E-02
-9.925E-02  9.894E-01 -9.429E-02
  2  4.1213E+01
  3  4.7271E+01
 20  2.5544E+02
 1.577E-23 -1.667E-01 -5.523E-01
-3.418E-02  1.408E-02  7.115E-01
 1.727E-01  7.750E-03  6.480E-01
 1.775E-01 -1.153E-01 -7.460E-01
 2.780E-02 -8.840E-02 -1.000E+00
```

Figure 2. Vibration problem output.

The buckling problem uses precisely the same data except that the second line (card group 1a) now indicates a buckling problem in which the effects of shear deformation are to be considered. The output is shown in Figure 3.

```
    1
 0.000E+00  0.000E+00  1.000E+00  3.000E-01  8.500E-01  0.000E+00
  4  1.0000E+01  1.0000E+06   9   1   6   5
   -1   2   3 -20
  1   0   1   1   2   3   1   3   4   2   4   4   2   5   6
  2   3   5   2   0   2   3   5   4   4   5   5
    1   1
 5.000E+06  9.000E+08  3.500E+01  0.000E+00  0.000E+00 -3.000E+00
 5.000E+06  9.000E+08  3.500E+01  1.250E+06 -4.000E+00  0.000E+00
 5.000E+06  9.000E+08  3.500E+01 -1.000E+06  4.000E+00 -3.000E+00
 5.000E+06  9.000E+08  3.500E+01  1.250E+06  4.000E+00  0.000E+00
 5.000E+06  9.000E+08  3.500E+01  0.000E+00  0.000E+00  3.000E+00
 5.000E+06  9.000E+08  3.500E+01 -1.000E+06  4.000E+00  3.000E+00
  1  5.3652E+00
 7.821E-25 -8.173E-04 -3.687E-01
 1.865E-03 -4.113E-03 -2.316E-01
-6.690E-03 -3.720E-03  1.000E+00
 1.600E-03 -1.059E-04  4.731E-01
-2.332E-03 -9.567E-04 -4.887E-01
  2  6.4428E+00
  3  8.1150E+00
 20  6.0739E+01
 3.198E-23 -2.726E-02 -5.386E-01
-1.343E-01 -4.296E-01 -2.254E-01
 5.595E-02 -4.280E-01 -7.921E-02
-1.299E-01 -1.000E+00 -2.351E-01
 1.093E-01 -9.627E-01  3.797E-01
```

Figure 3. Buckling problem output.

CONCLUSIONS

The Summary may also be read as a statement of the conclusions.

REFERENCES

Akesson, B.A. (1976) PFVIBAT - A computer program for plane frame vibration analysis by an exact method. *Int. J. Num. Meth. Eng.*, 10:1221-1231.

Blaszkowiak, S. and Kaczkowski, Z. (1966) *Iterative Methods in Structural Analysis*. Pergamon Press, Oxford.

Cowper, G.R. (1966) The shear coefficient in Timoshenko's beam theory. *J. App. Mech., Trans. ASME.*, 33:335-339.

Howson, W.P. (1979) A compact method for computing the eigenvalues and eigenvectors of plane frames. *Adv. Eng. Software*, 1(4):181-190.

Howson, W.P. and Williams, F.W. (1973) Natural frequencies of frames with axially loaded Timoshenko members. *J. Sound Vib.*, 26(4):503-515.

Lightfoot, E. (1980) Exact straight-line elements. *J. Strain Anal.*, 15(2):89-96.

Williams, F.W. (1981) Stability functions and frame instability - a fresh approach. *Int. J. Mech. Sci.*, 23(12):715-722.

THE USE OF MICRO-COMPUTERS IN THE ELASTIC ANALYSIS OF MULTI-CELL STRUCTURES

D.K. Pugh, R. Delpak and H.R. Evans
Polytechnic of Wales University College, Cardiff

ABSTRACT

The paper describes recent developments in the analysis of multi-cellular girders wherein the complex structure is idealized as a simple grillage. After outlining the theory and the general structure of the program that has been developed for solution on a microcomputer, results for typical structures are considered and compared. It is concluded that the program should prove to be a valuable design tool, particularly for the repeated analyses required to establish optimum structural proportions during the early stages of design.

INTRODUCTION

Multi-cellular structures are constructed from thin plates and comprise of two flange plates separated by a series of orthogonally intersecting webs. They combine a high bending and torsional stiffness, resulting in a construction that possesses strength and lightness. Such an efficient structural form has found wide application in the aircraft and shipbuilding industries and there are a number of examples of use in Civil Engineering, e.g. in box girder bridge decks and in dock gates.

Because of the complexity of the structural form and the variety of applied loading and support conditions that must be taken into account, the analysis of multi-cellular structures is difficult. In particular, it is difficult to establish an idealisation that can yield results of good accuracy without requiring a level of computational effort that is unrealistic at the preliminary design stage.

The availability of digital computers together with the development of the finite element method has proved to be a major step forward in the accurate analysis of these structures. However, to obtain an accurate solution, a large number of finite elements must be taken in the idealisation and the corresponding processing time is invariably high. The engineer is thus faced

with a dilemmaduring the early stages of design in trying to balance the need to study several alternatives against the analytical costs involved.

One method by which the required computational effort can be reduced to an acceptable amount is by idealising the cellular structure as a grillage. The validity of this approach was shown by Evans and Shanmugam in 1979 and one of the objects of this present paper is to discuss the conversion of the grillage program from a main frame computer to a microcomputer. In addition, several refinements of the original program, substantiated by the principles of elastomechanics, will be described and the application of the program to several practical problems will be illustrated.

GRILLAGE IDEALISATION AND ANALYSIS

A grillage is a structure consisting of orthogonally intersecting beam members. A typical beam element is assumed to have three degrees of freedom at each end, as shown in Fig. 1. These consist of two rotations about mutually perpendicular in-plane axes, and a transverse displacement; the resulting 6 x 6 element stiffness matrix is also shown in Fig. 1. Once such a matrix has been established for each grillage element the analysis of the structure can proceed according to the standard matrix stiffness method.

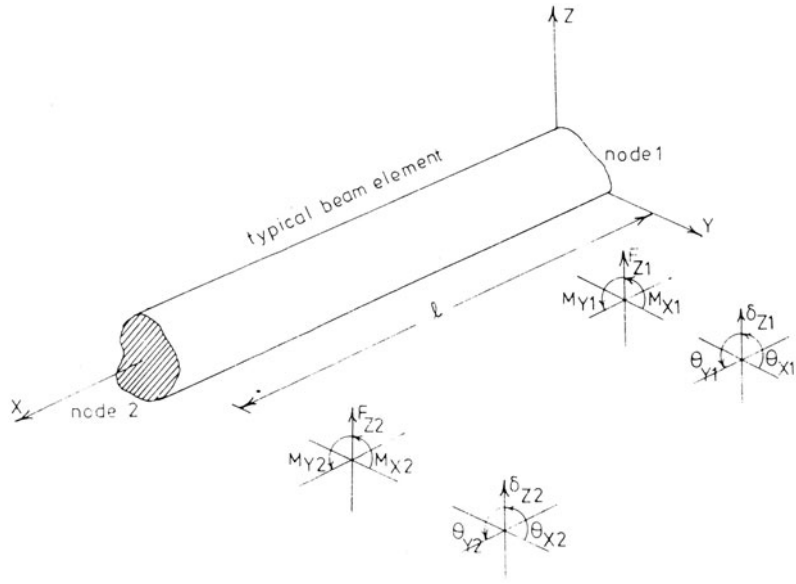
An idealisation of a typical multi-cellular structure as a grillage is shown in Fig. 2**c**. In this case, the centre lines of the equivalent grillage members coincide with the webs of the actual structure and each idealised member is an I-beam consisting of a web plate and an assumed effective width of flange.

There are, however, certain factors that must be considered before a continuous plated structure can be idealised by such an assemblage of discrete beam members. The four most important factors will now be discussed individually.

Shear Lag

Shear lag effects arise from the in-plane shear deformations that can develop in wide flanges. As shown in Fig. 2**a**, they lead to a non-uniform distribution of longitudinal stress and an amplification of the stresses developed close to the flange/web junction; the neglect of shear lag effects can thus lead to an underestimation of stress values.

Shear lag can be taken into account by adopting an effective width concept where the actual flange width is replaced by a reduced "effective width" over which the stress is considered to be uniformly distributed, as shown in Fig. 2 **a**. The extent to which the width must be reduced can be determined conveniently from empirical values established by Moffat and Dowling in 1975 on the basis of a comprehensive finite element study. These empirical values are now included in the recently published design code for steel bridges.



$$[K]_e = \begin{bmatrix}
\frac{12EI}{\ell^3}\frac{1}{1+2C} & \frac{6EI}{\ell^2}\frac{1}{1+2C} & 0 & -\frac{12EI}{\ell^3}\frac{1}{1+2C} & \frac{6EI}{\ell^2}\frac{1}{1+2C} & 0 \\
\frac{6EI}{\ell^2}\frac{1}{1+2C} & \frac{4EI}{\ell}\frac{1+\frac{C}{2}}{1+2C} & 0 & -\frac{6EI}{\ell^2}\frac{1}{1+2C} & \frac{2EI}{\ell}\frac{1-C}{1+2C} & 0 \\
0 & 0 & \frac{GJ}{\ell} & 0 & 0 & -\frac{GJ}{\ell} \\
-\frac{12EI}{\ell^3}\frac{1}{1+2C} & -\frac{6EI}{\ell^2}\frac{1}{1+2C} & 0 & \frac{12EI}{\ell^3}\frac{1}{1+2C} & -\frac{6EI}{\ell^2}\frac{1}{1+2C} & 0 \\
\frac{6EI}{\ell^2}\frac{1}{1+2C} & \frac{2EI}{\ell}\frac{1-C}{1+2C} & 0 & -\frac{6EI}{\ell^2}\frac{1}{1+2C} & \frac{4EI}{\ell}\frac{1+\frac{C}{2}}{1+2C} & 0 \\
0 & 0 & -\frac{GJ}{\ell} & 0 & 0 & \frac{GJ}{\ell}
\end{bmatrix}$$

Where I = second moment of area of the element cross-section

ℓ = length of element

J = polar second moment of area of the element cross-section

E = modulus of elasticity

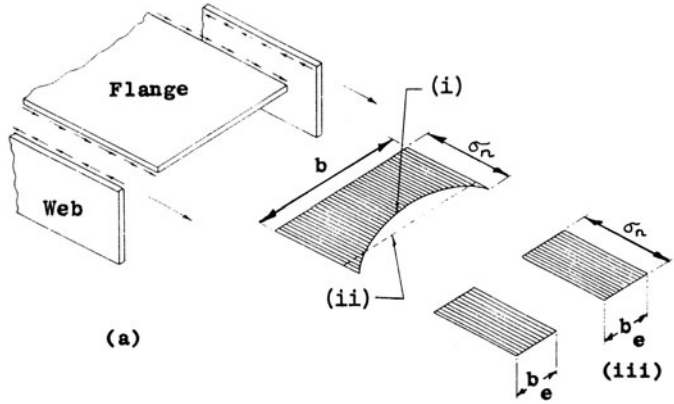
G = modulus of rigidity

$C = \frac{6EI}{GA_S\ell^2} = \frac{12I(1+\nu)}{A_S\ell^2}$

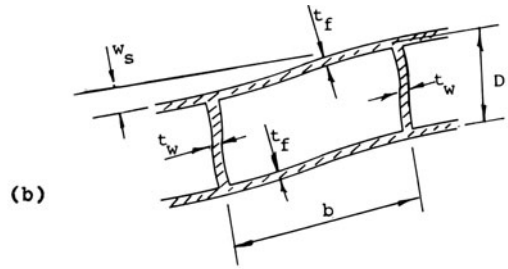
A_S = effective shear area of the element cross-section

ν = Poisson's ratio

Figure 1. Typical grillage beam element and element stiffness matrix



(i) Non-uniform flange stress due to shear lag
(ii) Stress distribution from simple beam theory
(iii) Assumed effective breadth of flange



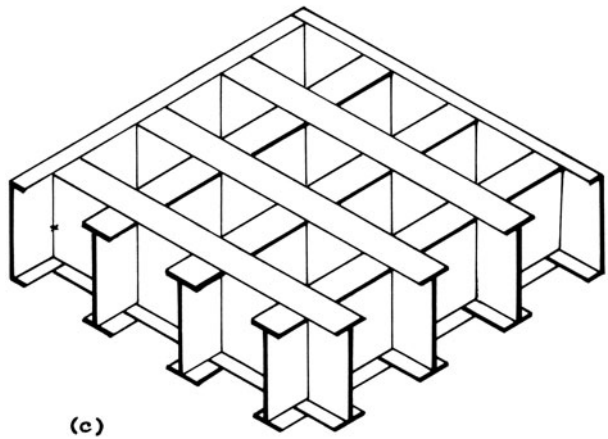


Figure 2. (a) Shear lag effects in flange plates; (b) cell distortion; (c) grillage idealization of typical multi-cell structure

By using these empirical values, the effective width of flange plate assumed to be acting with each web can be determined and the set of intersecting I-beams constituting the idealised grillage can be established.

Torsion

Because a multi-cellular structure comprises a series of closed cells, it is very stiff in torsion and adequate torsional stiffness must be specified in the grillage idealisation. The torsional stiffness of a multi-cell cross-section may be established by considering the shear flows in the walls of the cells. Such a procedure was described by Kollbrunner and Basler in 1969 and on this basis, the torsional constant J for the complete cross-section may be obtained as :

$$J = 4 \sum_{i=1}^{n} A_i \, q_i \qquad (1)$$

where n is the total number of cells, i denotes a typical cell, A_i is the area enclosed by the ith cell and q_i is the shear flow in the ith cell.

The value of the torsional constant J obtained using equation 1, for the typical 3-cell cross-section shown in Fig. 4 is 41.9 x $10^6 cm^4$. An analysis of the cross-section indicates that the shear flow in the internal webs is low in relation to the shear flow in the flange plates and external webs. If the shear flow in the internal webs is assumed to be zero, then the torsional constant for the complete cross-section is obtained as :

$$J = \frac{4 B^2 D^2}{2\left[\frac{B}{t_f} + \frac{D}{t_w}\right]} \qquad (2)$$

where B and D represent the overall width and depth of the section, respectively, and t_f and t_w indicate the flange and web plate thickness.

The simplified expression of equation 2 gives a J value of 41.6 x $10^6 cm^4$ for the cross-section in Fig. 4, agreeing closely with the value obtained from equation 1. This expression is, thus, used to determine the transverse torsional stiffness of the structure for the grillage idealisation.

The longitudinal section of the structure may also be considered as a closed thin-walled section and the longitudinal torsional stiffness may then be determined using an expression similar to equation 2.

Because the girder is treated as an equivalent orthotropic plate in the grillage idealisation, the overall torsional rigidity will contain the contributions from the two orthogonal directions. If the full values of the torsional constants for the two directions, calculated according to equation 2, are used, the overall torsional stiffness of the girder will be overestimated. It was proposed by Cusens and Parma in 1975 that the appropriate torsional constant to use in each direction was equal to half the value obtained from equation 2. Such a

procedure is followed in the present program so that half the transverse torsional stiffness of the girder is distributed equally to all longitudinal grillage beams whilst half the longitudinal torsional stiffness is distributed equally to the transverse beams.

Distortion

In thin-walled cellular structures without transverse diaphragms, substantial distortion of the cross-section may occur under certain loading conditions, as shown in Fig. 2b. Such behaviour cannot be reproduced precisely in a grillage beam but an approximation can be made by giving the transverse grillage beams a low shear stiffness.

The distortional behaviour may be compared to the deformation of a Vierendeel frame and the low shear stiffness may be chosen so that when the grillage beam and the Vierendeel frame are subjected to the same loading pattern, they experience similar distortions. A value of effective shear area, A_s, is calculated for the grillage beam such that the shear deflection and bending deflection of the beam are equal to the deflection of the Vierendeel frame at corresponding positions.

The Vierendeel frame may be analysed by standard methods. A convenient approximation, proposed by Hambly in 1976, is obtained by assuming that the shear force is shared between top and bottom flange plates in proportion to their individual flexural stiffnesses and that there are points of contraflexure midway between the webs. This approximation results in the following expression for the equivalent shear area of the transverse grillage beams :

$$A_s = \frac{2t_f^3}{b^2}\left[\frac{t_w^3 b}{t_w^3 b + 2Dt_f^3}\right]\frac{E}{G} \quad \text{per unit width} \qquad (3)$$

where the dimensions are as defined in Fig. 2b. This expression has been incorporated in the grillage program.

Transverse Bending

In the absence of in-span diaphragms, the bending stiffness to be apportioned to the transverse grillage beams is determined from the full bending stiffness of the top and bottom flange plates of the girder. When a diaphragm is positioned within the span, then its contribution to the bending stiffness of the grillage beam is included.

PROGRAM

The program has been developed to run on a 32K Commodore Pet micro-computer. It incorporates the following important features:

a) the minimum of data preparation is required.
b) the program is structured such that data may be stored within the memory of the micro-computer, without the need for disk storage. The reliance on hardware is kept to a minimum.

POINT LOADS AT MIDSPAN OVER ALL WEBS

MATERIAL PROPERTIES

POISSON'S RATIO = .3

PROP NO.	E	I	J	A
1	21000.0000	3037500.	7899429.	180.000
2	21000.0000	2269350.	5207429.	180.000
3	21000.0000	4201200.	5207429.	180.000
4	21000.0000	5400000.	7899429.	.020

JOINT LOADING

JOINT NO.	PZ	MX	MY
9	- 300.000	+ .000	+ .000
10	- 600.000	+ .000	+ .000
11	- 600.000	+ .000	+ .000
12	- 300.000	+ .000	+ .000

JOINT DISPLACEMENTS

JT NO.	Z-DISPL	X-ROTN	Y-ROTN
1	+ .000000	+ .000000	+ .001054
2	+ .000000	+ .000000	+ .001062
3	+ .000000	+ .000000	+ .001062
4	+ .000000	+ .000000	+ .001054
5	- .426616	- .000000	+ .000788
6	- .472850	- .000000	+ .000798
7	- .472850	+ .000000	+ .000798
8	- .426616	+ .000000	+ .000788
9	- .642322	- .000000	+ .000000
10	- .733193	- .000000	- .000000
11	- .733193	+ .000000	- .000000
12	- .642322	+ .000000	+ .000000
13	- .426616	- .000000	- .000788

MEMBER FORCES

MBR.	JT.	PZ	MX	MY
4	1	+ 150.049	+ 1.779	+ 1717.567
4	5	- 150.049	- 1.779	- 61737.309
5	2	+ 299.950	+ .913	- 1717.567
5	6	- 299.950	- .913	-118262.689
6	3	+ 299.950	- .913	- 1717.567
6	7	- 299.950	+ .913	-118262.689
7	4	+ 150.049	- 1.779	+ 1717.567
7	8	- 150.049	+ 1.779	- 61737.308
8	5	+ .024	- 2164.765	- .619
8	6	- .024	+ 2164.765	- 6.848
9	6	+ .000	+ .000	+ 6.565
9	7	- .000	+ .000	- 6.565
10	7	- .024	+ 2164.765	+ 6.848
10	8	+ .024	- 2164.765	+ .619
11	5	+ 150.024	+ 1.160	+ 63902.074
11	9	- 150.024	- 1.160	-123911.859
12	6	+ 299.975	+ .630	+116097.924
12	10	- 299.975	- .630	-236088.138

LONGITUDINAL STRESSES

JT NO.	STRESS N/MM2	JT NO.	STRESS N/MM2
5	19.94	6	20.03
7	20.03	8	19.94
9	52.17	10	54.90
11	54.90	12	52.17
13	19.94	14	20.03
15	20.03	16	19.94

Figure 3. Typical output

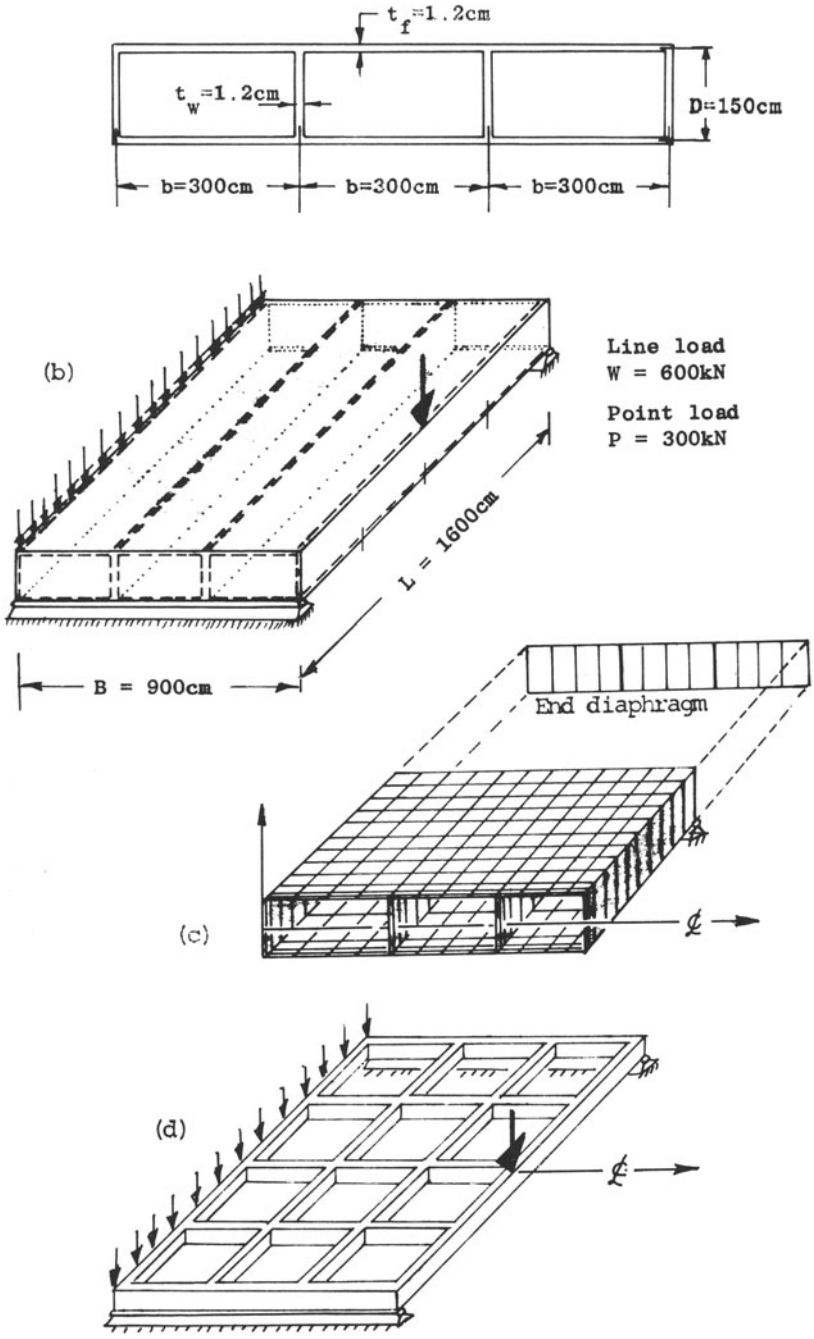


Figure 4. (a) Dimensions of cross-section; (b) details of three cell structure; (c) finite element idealization; grillage idealization

c) the output is presented in a simple format which can be easily interpreted, see Fig. 3 .
d) two possible methods are available for the input of data - from the keyboard or using the READ/DATA facility in PET BASIC.

Program Structure

The program may be divided into 3 sub-sections :
a) Input of data and calculation of structure and loading details.
b) Establishment of the structural stiffness matrix and load vector, solution and calculation of joint displacements.
c) Calculation of member forces and bending stresses.

The solution routine incorporated in the program takes advantage of the banded symmetrical nature of the stiffness matrix. Only the upper half of the band is stored and a Gaussian elimination technique is employed to save both storage and computer time. Backsubstitution through the modified stiffness matrix produces the joint displacements. The member forces are then calculated from the element stiffness matrix and appropriate displacements in the usual manner.

Data Input

The input routine has been developed to minimize the amount of data that has to be entered by the user. The joint co-ordinates and element connectivity are determined from the specified values of span, width, number of cells and number of transverse grillage beams, the mesh being generated within the program.

The material properties are required and the thickness of the flange, web and diaphragm plates must be specified, together with the shear lag effective width factors. The program then determines the values of second moment of area, torsional constant and shear area to be allocated to each grillage member using the appropriate equations discussed earlier. Applied loading may be specified as either member or joint loads.

TYPICAL RESULTS

The typical 3-cell girder shown in Fig. 4 was analysed under four different loading conditions and the results obtained from the grillage analysis were compared to those obtained from the finite element method. In all cases the girder was considered to be simply supported at both ends, representing typical support conditions for a bridge deck.

The finite element results for comparison purposes were obtained from a standard computer program QUEST; the accuracy of this program in the analysis of box girders is well established. A convergence study was carried out to determine the finite element idealisation required and the mesh shown in Fig. 4, consisting of 348 elements on the half-structure, was found to be satisfactory, the mesh is seen to be graded in the region of the mid-span point loads. The corresponding grillage idealisation is also shown in Fig. 4.

The results obtained are summarized in Figs. 5 and 6. Figure 5 shows the calculated longitudinal stress values at the mid- and quarter-span cross-sections for each of the four loading conditions considered; corresponding values of vertical deflections are plotted in Fig. 6.

In the first two loading conditions considered, point loads were applied at mid-span whereas line loads were applied in the two remaining cases. All loads were applied above the webs with the transverse positions of the loads being varied to produce different amounts of distortion in the cross-section.

The stresses plotted in Figs. 5 for the two point loading conditions show the very significant shear lag effect at the mid-span section under the load; at quarter span, no shear lag is apparent. The shear lag effect at mid-span is less marked for the line loading cases although, in this case, shear lag is also apparent at $\frac{1}{4}$-span.

Figure 6 shows little cross-sectional distortion to have been produced by either the point or line loads when they were applied to all webs. However, as expected, the cases in which point loads were applied to the two inner webs only and when line loads were applied to the two outer webs produced considerable distortion.

In each case, the results obtained from the grillage and finite element solutions are in very good agreement. The grillage values, plotted as points, are seen to lie close to the finite element curves and certainly exhibit a degree of accuracy that would be acceptable at a preliminary design stage.

It is of interest to compare the computational effort required by the two solutions. The finite element program was run on a DEC system main frame computer and the mesh shown in Fig. 4 required the consideration of a total of 2,028 degrees of freedom; the solution for one loading case took about 15 minutes of computer time. In sharp contrast, the grillage solution was carried out on a 32K Commodore Pet micro-computer and required the consideration of only 60 degrees of freedom. The solution on the micro-computer took about 10 minutes.

One difficulty in trying to establish the accuracy of the grillage approach is the expense incurred in carrying out the finite element analyses for comparison. This usually limits the number of cells that can be considered. Very little experimental information is available for comparison, but Fig. 7 shows details of a 60-cell, $\frac{1}{4}$-scale model of a ship's hull tested by Mitchell in 1976. This model was very complex, having different sized openings in the longitudinal and transverse webs as shown.

Experimental values were only given for the deflections and Fig. 7 compares the experimental deflections to those predicted by the grillage method. The comparison is very satisfactory, particularly when it is noted that a slight movement was recorded at the edge of the experimental model; this edge was

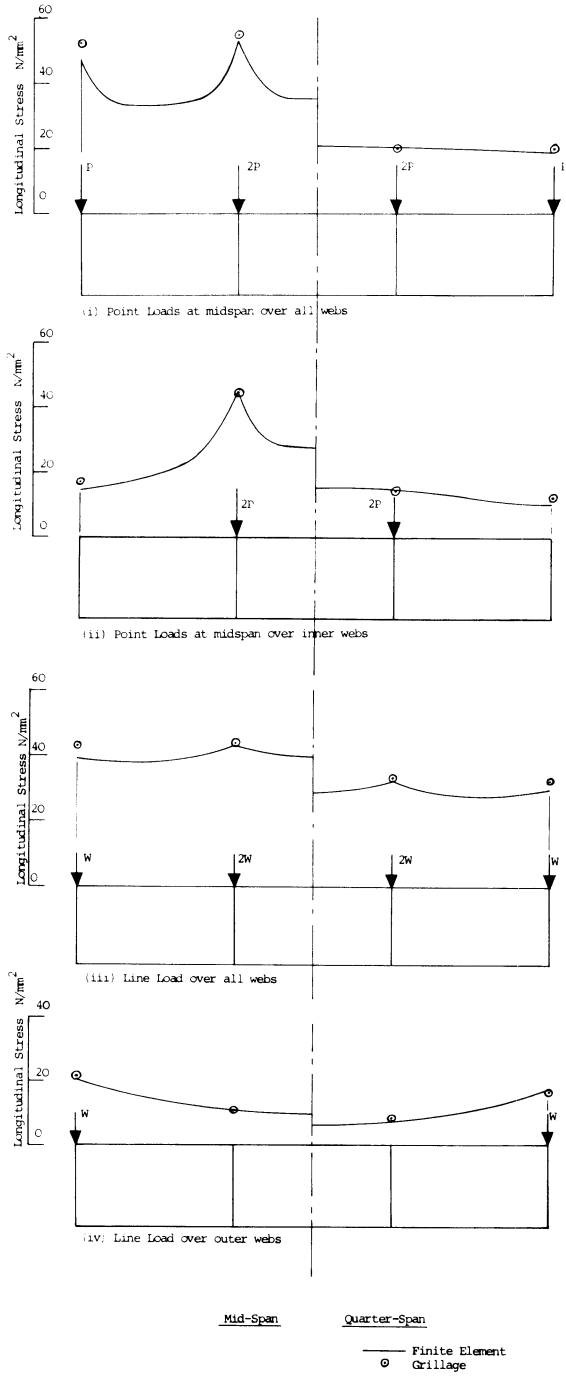


Figure 5. Variation of longitudinal stress at midspan and quarter span

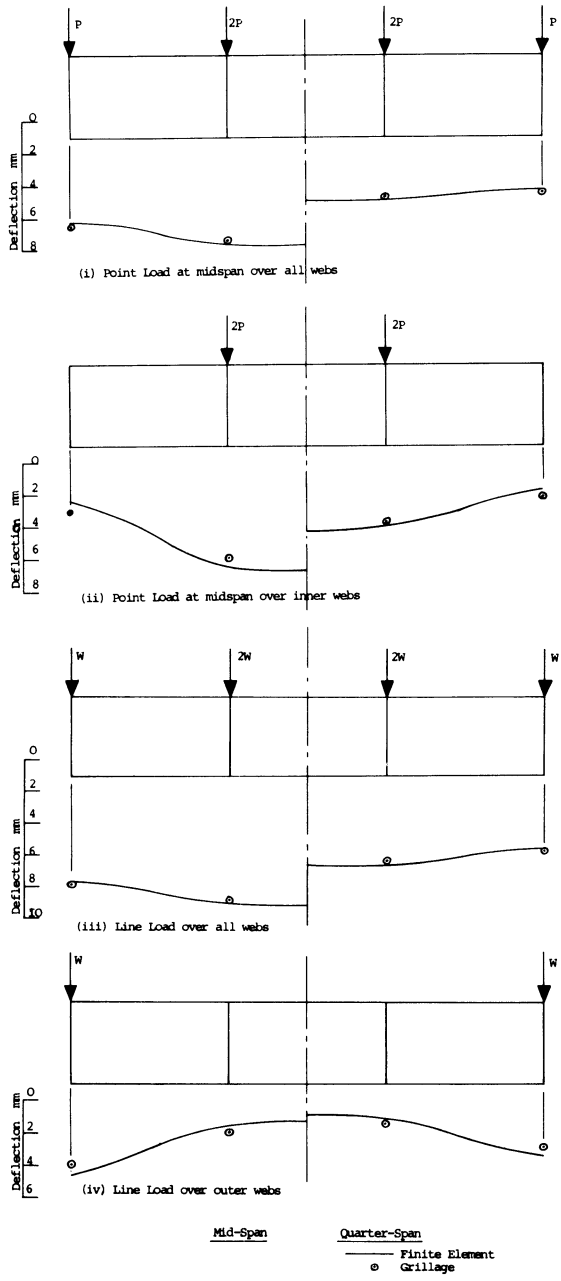
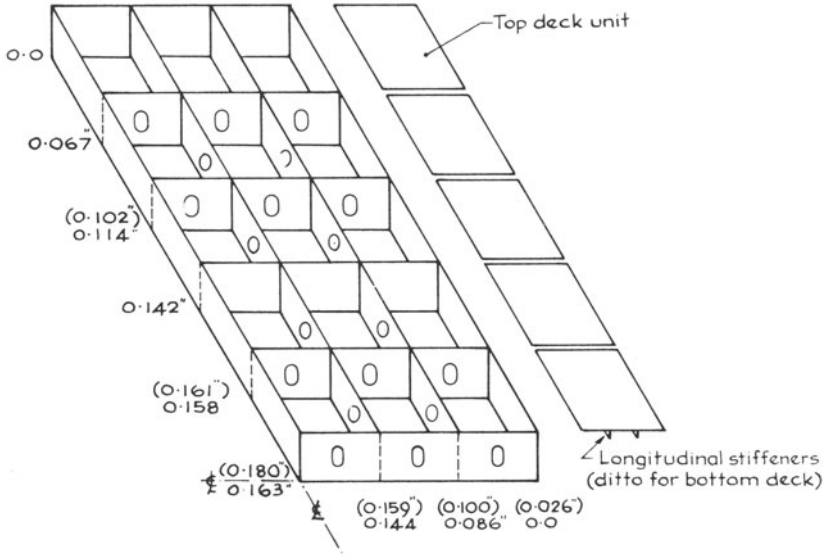


Figure 6. Variation of vertical deflection at midspan and quarter span



Experimental values bracketed
Calculated values unbracketed

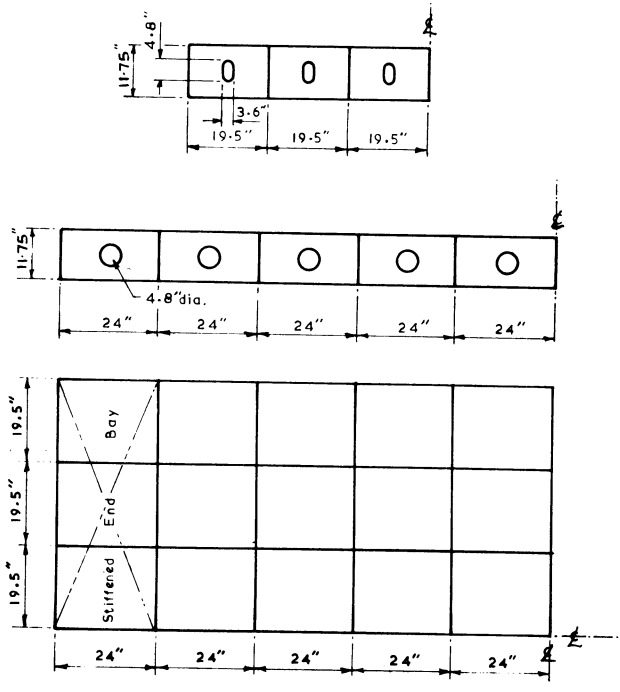


Figure 7. Details of quarter scale test model of ship double bottom (Mitchell)

nominally simply supported and specified as such in the grillage analysis.

CONCLUSION

Although only a limited number of results have been presented in this paper, it is apparent that the grillage method provides a fast, economic and reliable method for the analysis of multi-cellular structures. The results have shown its application to both bridge and ship-type structures and have thus illustrated its versatility. The micro-computer program that has been developed should prove to be a valuable design tool, particularly for the repeated analyses required to establish optimum structural proportions during the early stages of design.

REFERENCES

Cusens, A.R. and Parma, R.P. (1975). Bridge Deck Analysis. John Wiley, London.

Evans, H.R. and Shanmugam, N.E. (1979). An Approximate Grillage Approach to the Analysis of Cellular Structures. Proc.Instn. Civ.Engrs., Part 2, 67, Mar., 133-154.

Hambly, E.C. (1976). Bridge Deck Behaviour, Chapman and Hall, London.

Kollbrunner, C.F. and Basler, K. (1969). Torsion in Structures, translated from German by E.C. Glauser, Springer-Verlag, Berlin.

Mitchell, G.C. (1976). Structural Test on a $\frac{1}{4}$-scale Ship's Double Bottom Model. Naval Construction Research Establishment, Dunfermline. Report R640.

Moffatt, K.R. and Dowling, P.J. (1975). Shear Lag in Steel Box Girder Bridges. Struct. Engr., 53, No. 10, Oct., 439-448.

ACKNOWLEDGEMENTS

The Authors wish to record their appreciation to the Heads of the Civil Engineering Departments at both academic institutions for making the departmental facilities available.

MABEL - A MICRO-COMPUTER PROGRAM FOR THE MODAL ANALYSIS OF BUILDINGS UNDER EARTHQUAKE LOADS

Harold E. Santo

CMEST - Centro de Mecânica e Engenharia Estruturais da Universidade Técnica de Lisboa,
Av. Rovisco Pais, 1096 Lisboa Codex, Portugal

SUMMARY

A microcomputer program which performs the modal analysis of plane frames subjected to earthquake loads, entitled MABEL (Modal Analysis of Buildings under Earthquake Loads), part of a "CAI in Structural Analysis" Project, is described. The role of computer assisted instruction and computer graphics in engineering education is assessed and thoroughly discussed. It is argued that computer graphics is helping to bridge the gap between academic training and professional requirements, through its abilility to provide an instant visual guide to intuition, a factor overlooked in recent times.

INTRODUCTION

Unlike computer graphics, computer assisted instruction has not really yet come of age. In spite of elaborate projects like PLATO and TICCIT, several factors prevented its wider adoption and greater development throughout the sixties and seventies, the two main reasons being, perhaps, expensive hardware and student/user interface shortcomings (the latter best exemplified by Luehrman (1972) and Lawrence/Sheppard (1977), out of CAI and CAD fronts). Another reason for CAI lack of success is that computers and computing evolved faster than educators and schools could ever keep up with (Years ago there used to be a debate on whether electronic calculators should be allowed in exams. Today the same seems to be happening with microcomputers ("Computer Bust. Processors Banned from Exams", Time, April 19, 1982)).

In the last few years, however, the gathered experience and lessons learned with past mistakes, the steady decline of hardware cost and the appearance of low-priced microcomputers and equipment, have brought new hopes and possibilities to the implementation of computer based curricula at all levels of

education. Recent articles sample this undoubtful reality (Nievergelt (1980) and Juricic/Barr (1982), also from CAI and CAD fields).

Nevertheless, it should be pointed out that economic drawbacks are still a worldwide universities' handicap (Scardina, et al (1980)) so, in most cases, the arrival of the micros with graphics, meant more than a mere added benefit: it brought the only viable alternative to computer aided education.

But, while the need to incorporate computer graphics and CAI material into engineering courses is an unquestionable fact nowadays, very few schools have already begun to undertake such a task. In the case of structural analysis, advanced efforts appear to remain, unfortunately, just a valuable source of inspiration (Kamel (1978), O'Rourke, et al (1979), Gattass, et al (1981)).

The overall point is, in sum, that engineering education has to change, taking advantage of all recent technological advances. CAI - probably the only major breakthrough since the advent of the slate - will have to spread if engineering schools are to be in tune with the times, so every action towards the application of computers and computer graphics in education, with whatever tools available, should be highly encouraged.

THE CAI PROJECT

Drawing on the ideas outlined above, a "CAI in structural Analysis" Project is being carried out by the author at Instituto Superior Técnico (Technical University of Lisbon's Engineering Institute). A project of this kind is, of course, no attempt to substitute hand-calculation or experimental methods but to add to them. The underlied philosophy follows the basic notion that, with a computer graphic aid, a student can gain a "feeling" and train the intuition (a must for a structural engineer) better and faster than by any other means. The possibility to try a very great number of cases and promptly visualize the structural behavior seems to be the greatest ever educational achievement in the field.

It is assumed, for the sake of these notes, that the project is developed in close relationship with the entire engineering curriculum, supporting and complementing the structural analysis courses. Also that the students have already acquired a minimum background in computing and computer graphics (the latter condition is, in fact, guaranteed, for an introductory computer graphics course has been successfully implemented, at the civil engineering freshman level, within the basic engineering graphics discipline, an accomplishment which, incidentally, the author is responsible for (Santo (1982)). Other basic conditions supposed to be met are the existence of a CG lab, with adequate student/utilization-

-time ratio, and continuing assistance of the faculty.

The general characteristics of the whole project can thus be enumerated:

. it has a problem-oriented language structure
. standard exercises, covering the whole spectrum of the S.A. courses, are featured
. extensive graphical (2D/3D) capability, with use of tablet, mini-plotter, screen displays (with associated hard-copy unit), is provided
. numerical output is large and user-controlled
. the programming of applications is straightforward
. individual modules are set in a standard fashion, anticipating professional procedures.

The introduction of a "CAD in Civil Engineering" course, right before or parallel to the S.A. disciplines, would complete the updating of the advanced undergraduate program. It is a feasible fundamental idea, which is still, unhappily, not generally pursued as should be.

The term CAI - taken for granted herein - though still not a commonplace at engineering undergraduate levels, let alone graduate, best defines the primary topic of this paper: the utilization of computers to assist and enhance the teaching and learning processes (the acronym adopted in Britain - CAL, for computer assisted learning - by the way, has a slight different meaning but, in reality, these designations and their definitions are not totally agreed upon. CAD is another example. A strong need for standardization here!).

THE MICROCOMPUTER PROGRAM MABEL

The microcomputer program MABEL (Modal Analysis of Buildings under Earthquake Loads) is part of the early stages of the CAI Project and, in its present version, performs the modal analysis of plane frames subjected to earthquake loads.

Its prominent aspects are

. speedy definition of the structural topology through a tablet
. immediate validation and correction of data
. interactive processing
. several conditions of analysis, depending on material, response spectrum, structure type
. analysis by the Portuguese Earthquake Code
. extensive numerical and graphical output.

Besides being a learning aid, the program can serve as a means to obtain preliminary dimensioning of structures in actual projects.

Theoretical Assumptions

The program incorporates the standard theory, found in the current literature (Hurty/Rubinstein (1964), Clough/Penzien (1975), Bathe/Wilson (1976)) and taught in IST's structural analysis courses.

It can be summarized as follows

. the equilibrium equation is of the form

$$\underset{\sim}{m}\,\ddot{\underset{\sim}{x}} + \underset{\sim}{c}\,\dot{\underset{\sim}{x}} + \underset{\sim}{k}\,\underset{\sim}{x} = -\underset{\sim}{m}\,\underset{\sim}{I}\,\ddot{x}_s$$

in which $\underset{\sim}{m}$ - mass matrix
$\underset{\sim}{c}$ - damping matrix
$\underset{\sim}{k}$ - stiffness matrix
$\underset{\sim}{I}$ - unitary vector
$\ddot{x}_s$ - acceleration of the soil
$\underset{\sim}{x}, \dot{\underset{\sim}{x}}, \ddot{\underset{\sim}{x}}$ - displacement, velocity and acceleration vectors of the lumped masses

. the frequencies and modes of vibration are computed from the solution of the homogeneous linear equation system (obtained by deleting, in the equilibrium equation, the damping and applied forces members) $|k - w_o^2\ m|\ \underset{\sim}{\phi} = \underset{\sim}{o}$
. the mode shapes are the eigenvectors of the $|m^{-1}\ k|$ matrix and the squared frequencies, w_o^2, are the eigenvalues.
. the participation factor and generalized mass for each mode j, are calculated respectively by

$$P_j = \sum_{i=1}^{n} m_i\ \phi_{ij} \qquad m_j = \sum_{i=1}^{n} m_i\ \phi_{ij}^2$$

. the nodal displacement for each mode j is obtained from

$$x_j = \frac{P_j}{m_j}\ \underset{\sim}{\phi}_j\ S_{dj}$$

S_{dj} - spectral displacement (function of period and relative damping)

$\frac{P_j}{m_j}$ - modal participation factor

. the displacement, velocity and acceleration matrices are calculated by the products $\underset{\sim}{X} = \underset{\sim}{x}\ \underset{\sim}{\phi}$, $\dot{\underset{\sim}{X}} = \underset{\sim}{x}\ w_o\ \underset{\sim}{\phi}$, $\ddot{\underset{\sim}{X}} = \underset{\sim}{x}\ w_o^2\ \underset{\sim}{\phi}$.
. the final earthquake forces matrix is obtained by $\underset{\sim}{F} = \underset{\sim}{X}\ \underset{\sim}{k}$
. the maximum displacements, velocities, accelerations and earthquake forces, for each floor, are computed by the root-mean-square procedure
. the final forces are divided by the ductility factor to take into account, approximately, the nonlinear behaviour
. the seismic coefficients are calculated by $c = \frac{E.F.(max)}{g\ M}$.
. each mode has the same relative damping ratio ($o < d.r. < 0.2$)
. the relative displacements of consecutive floors correspond to the simple difference of maximum displacements.

Program Features

Stiffness matrix The stiffness matrix can be generated from two basic assumptions:

. no joint rotation
  the floors are considered infinitely rigid.
  the stiffness of a story i is then given by

$$k_i = \sum_{j=1}^{N} \frac{12\,E\,I_j}{h_i^3} \quad (N - n^{o} \text{ of columns})$$

. joints free to rotate
  in this case the condensed stiffness matrix, in terms of the horizontal displacements of the floors, is derived by the well-known expression $\underset{\sim}{k} = \underset{\sim}{k}_{11} - \underset{\sim}{k}_{12} \times \underset{\sim}{k}_{22}^{-1} \times \underset{\sim}{k}_{21}$.

The stiffness matrix can also be obtained by inversion of the flexibility matrix, which is input as data. This is to allow the case where the flex matrix is available, e.g., using another program, applying unitary forces at floor levels, taking into account axial strains.

Spectral Displacements Two cases can be considered:
. El Centro 1940
  an appropriate subroutine generates the data, built from the response spectrum presented in Clough (1970).
. Portuguese Earthquake Code
  another routine incorporates the Code's response spectra.

Number of Modes in the Analysis For each analysis several cases can be considered, with the inclusion of any desired number of modes in the computations.

Stresses and Graphical Output For each member, bending moments, shear and axial forces are produced. The graphical portion includes:
. modes of vibration
. maximum and relative displacements
. maximum velocities and accelerations
. maximum earthquake forces and seismic coefficients
. global bending moments and shearing forces.

Summary of Possible Cases In short, the program allows the
. study of the effects of joint rotation and axial strains
. comparison of the response of the structure due to an El Centro-type earthquake and the Portuguese Code requirements
. comparison of the response due to various ductility factors and damping ratios
. study of the effects of high order modes of vibration.

Data Structure and Output Sequence
The data required by the program consist of the following blocks:

. Title of the case
. Control parameters
  Number of stories, number of bays, Young modulus, damping ratio, ductility factor
  Type of matrix — no joint rotation / free rotation / flex. matrix as data
  Type of spectrum — El Centro / Portuguese Code
  Terrain index — five types of soil, as defined in the Code
. Properties of the members, frame topology and masses of the floors
. Number of cases and number of modes for each case.

The output comprises
. Input data
. Stiffness and flexibility matrices
. Frequencies, periods and normalized mode shapes
. Participation factors, generalized masses, spectral and modal displacements
. Displacement, velocity, acceleration and force matrices
. For each floor
  maximum and relative displacement
  maximum velocity, acceleration and earthquake forces
  seismic coefficient
. Bending moments, shearing and axial forces
. Graphical diagrams (as already quoted)

Future Developments

Although the program fulfills the principal objectives, it is still limited in scope. In a second phase, therefore, additional facilities will be included, such as

. other modal combination options (which will permit the comparison of various superposition criteria)
. inclusion of rotational, lateral and vertical spring supports (to take into account the foundation flexibility)
. consideration of shear walls.

This extension will form the basis of a complete tridimensional building analysis micro-computer package, in which each plane frame will be treated as a substructure. Torsional modes and varied earthquake directions will thus be handled properly. The latest Earthquake Code version will also be accounted for.

CONCLUSIONS

Overall guidelines for the introduction of computer graphics -oriented material in a structural analysis course were introduced.

The implied philosophy underneath the implementation of "CAI in Structural Analysis" also stems from the current urgency to close the gap between education and engineering practice, particularly with regard to the role of computers (a concern not totally overlooked lately - ASCE (1979)).

Structural analysis instruction has to be a reflection of the professional environment and needs, has to provide the background for virtual researchers and the preparation for would-be educators.

Carefully crafted CAI projects are certainly the ultimate contribution.

ACKNOWLEDGEMENTS

The support of CMEST in the preparation of this paper and development of the project is gratefully acknowledged. Mrs Sílvia Santos careful typing, student Vasco Ferreira micro--computer work and Mr. Jorge Fernandes drawings are very much thanked for.

REFERENCES

ASCE, Conf. on Elect. Comp. (1979) Are the Universities Educating Engineers Adequately for the Effective Use of Computers?, Panel Discussion

Bathe, K.J., and Wilson, E.L. (1976) Numerical Methods in Finite Element Analysis. Prentice-Hall, Inc.

Clough, R.W. (1970) Earthquake Response of Sructures. In "Earthquake Engineering", R.L. Wieger, editor. Prentice-Hall, Inc.

Clough, R.W. and Penzien, J. (1975) Dynamics of Structures. McGraw-Hill, Inc.

Gattass, M., et al (1981) Interactive Graphics Dynamic Analysis of Frames. ASCE Proc. - 2nd Speciality Conf. on Dyn. Resp. of Str.: 502-515.

Hurty, W.C. and Rubinstein, M.F. (1964) Dynamic of Structures. Prentice-Hall, Inc.

Juricic, D.,Barr, R.E. (1982) Graphics and CAD - A Systematic Approach. Mechanical Engineering, 104, 9: 48-53.

Kamel, H.A. (1978) Interactive Computer Graphics as a Tool in Teaching Structural Engineering. ASCE - J. Str. Div., 104, 8: 1299-1314.

Lawrence, D.J., and Sheppard, D.J. (1977) Computer Aided Design Education - The More the Education, the Less the Use. Proc. CAD ED Conf., IPC Press.

Luehrman, A.W. (1972) Should the Computer Teach the Student, or Vice-Versa? AFIPS Conf., 40: 407-410.

Nievergelt, J. (1980) A Pragmatic Introduction to Courseware Design. Computer, 13, 9: 7-21.

O'Rourke, M.J., et al (1979) Using Computer Graphics to Teach Structural Dynamics. ASCE - Proc. 2nd Conf.Civil Engng Educ., Vol I: 563

Santo, H.E. (1982) The Teaching and Application of Computer Graphics within an Engineering Graphics Environment.Submitted for presentation in the CAL' 83 Conference.
Scardina, J.T., et al (1980) Computer Graphics as a Teaching Aid. ASCE - Proc. 2nd. Conf. Comp. Civil Engrg.: 430-435.

APPENDIX

Case Study

The analysis of a ten-story, three-bay frame was carried out, under various conditions, by MABEL and a reliable program (ESTATISMAX from LNEC, the Portuguese National Civil Engineering Laboratory), with excellent agreement in the results (comparison not presented).

Figures 1,2 and 3 show comparative approximate curves for the typical cases indicated.

The remaining figures are examples of mini-plotter graphical output.

As complement, the following table, which illustrates the influence of higher modes, is included (the values refer to the top floor).

Percentage of the Total				
Nº of Modes Cons.	El Centro		Code	
	Disp.	Earth.Force	Disp.	Earth.Force
1	99.26	65.98	99.52	70.07
2	99.96	86.71	99.96	84.73
3	99.996	95.49	99.99	92.70
4	99.999	98.49	99.999	97.59
10	100.00	100.00	100.00	100.00

Calculation of the Seismic Coefficient by the Portuguese Code

The early draft (1975) used in the present version of MABEL, considers the following expression:

$$c = \alpha\, c_o \frac{\beta}{\mu}$$

where α - seismicity factor which depends on the seismic zone
c_o - coefficient which depends on the soil type and the natural fraquency
β - factor that introduces the influence of higher modes of vibration
μ - ductility factor

For the 10-story frame:

. $\alpha = 10$ (seismic zone A - Lisbon)

. The natural frequency is given by $f = \frac{20}{n}$ (n = number of stories)

$$f = \frac{20}{10} = 2.0$$

. For soil type II (current terrain) and $0.5 < f < 5.0$,

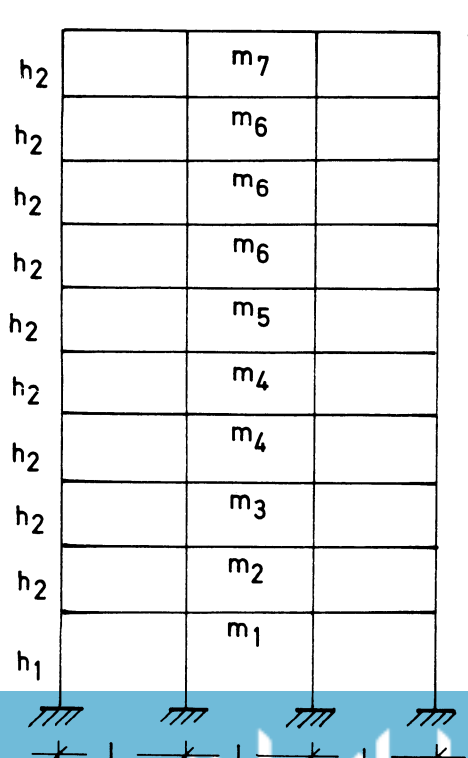
$$c_o = 0.16 \sqrt{f} = 0.226$$

. $\beta = 1.5 - \frac{f}{5} = 1.5 - \frac{2.0}{5} = 1.1$

(for a MDOF structure with $f < 2.5$ Hz)

. $\mu = 3.0$ (R/C structure)

. $c = 1.0 \times 0.226 \times \frac{1.1}{3.0} = 0.083$



$E = 3.4 \times 10^6$ t/m^2

Damp. Ratio = 0.05

Duct. Fact. = 3.0

$h_1 = 6$m $h_2 = 4$m $L = 8$m

$I_1^V = 0.03645 m^4$ $I_2^V = 0.0108 m^4$

$I_1^P = 0.3456\ m^4$

$I_2^P = 0.1458\ m^4$

$I_3^P = 0.0432\ m^4$

$m_1 = 26.08$ t sec^2 m^{-1}

$m_2 = 25.35$ "

$m_3 = 24.98$ "

$m_4 = 24.61$ "

$m_5 = 24.25$ "

$m_6 = 23.88$ "

$m_7 = 15.67$ "

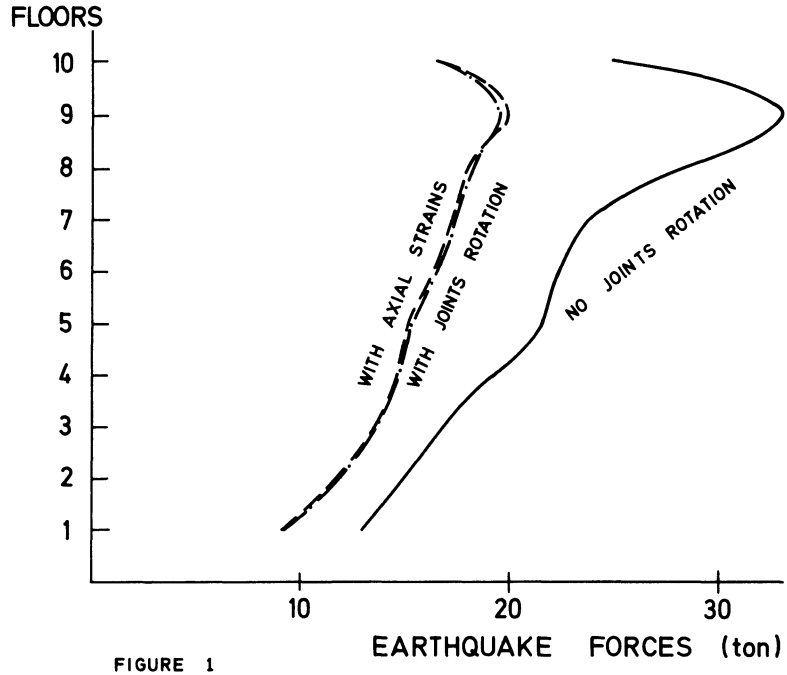


FIGURE 1

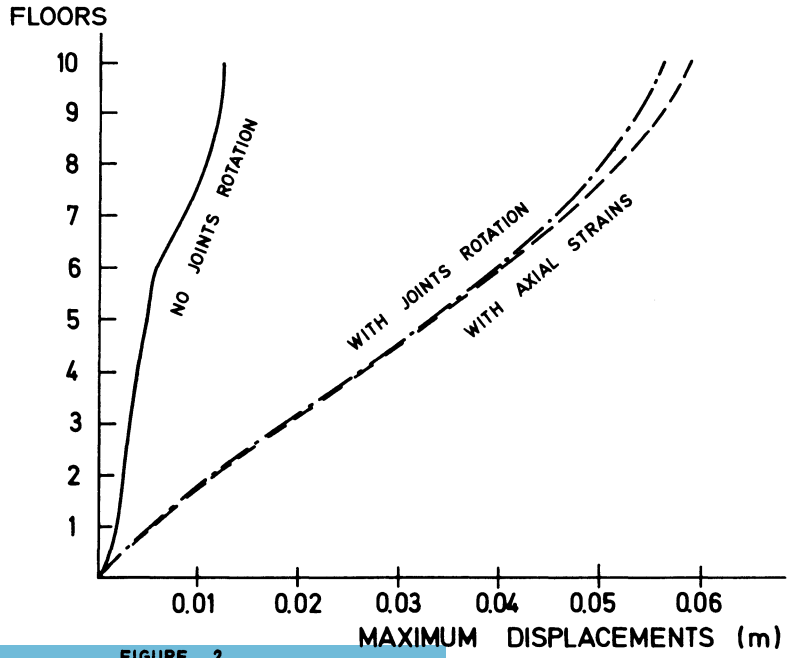


FIGURE 2

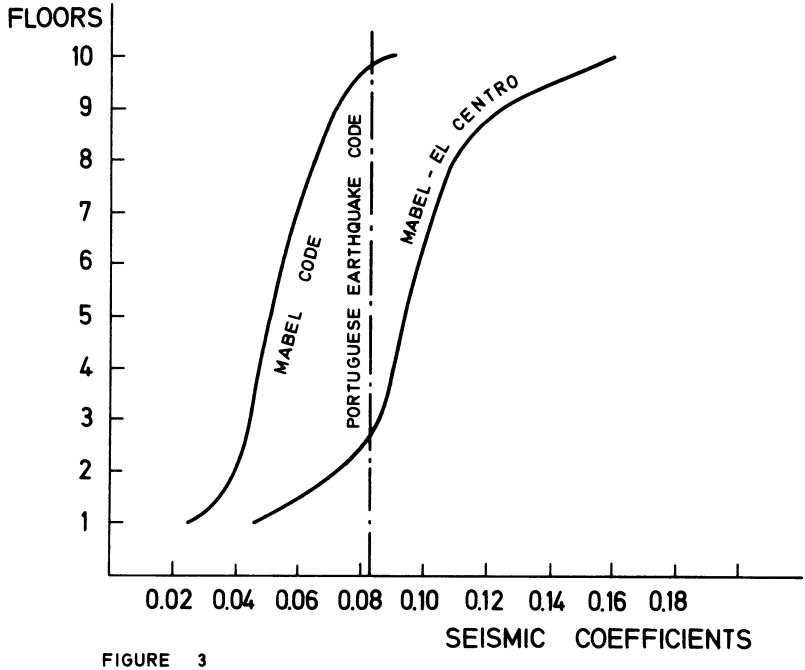
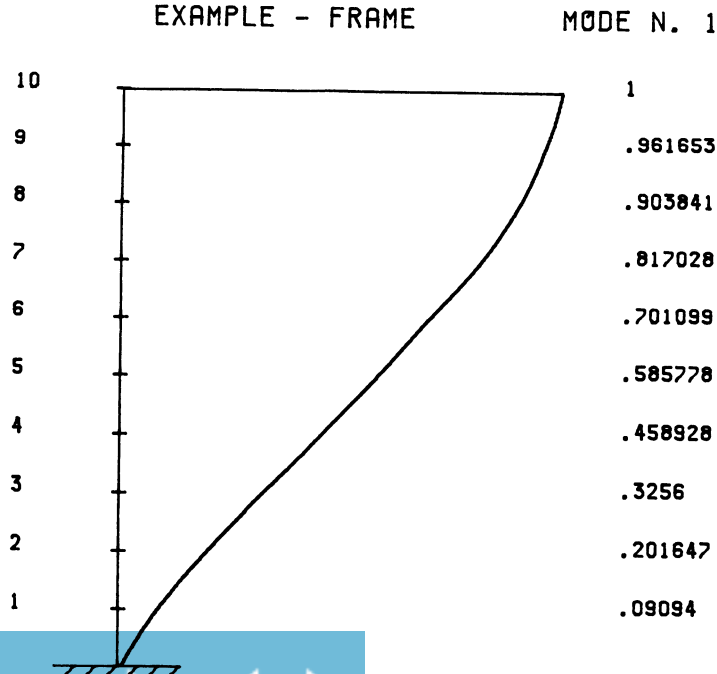
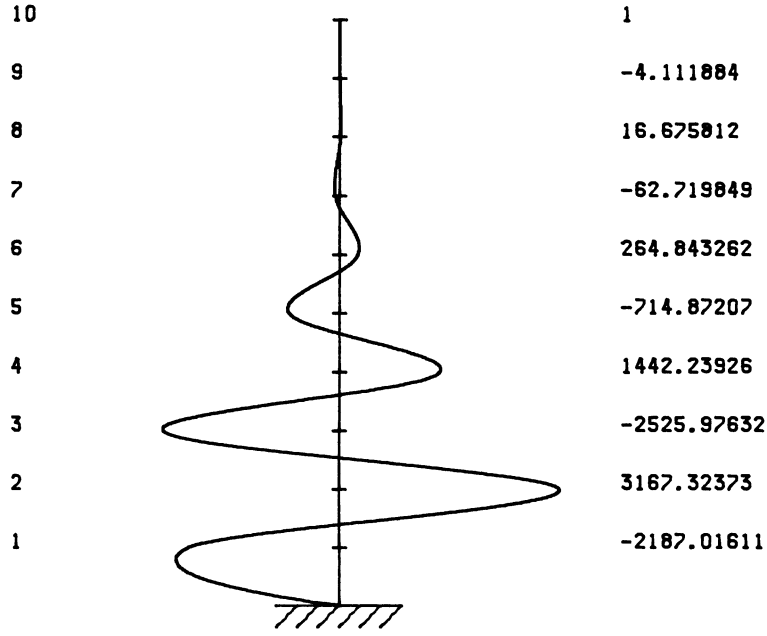


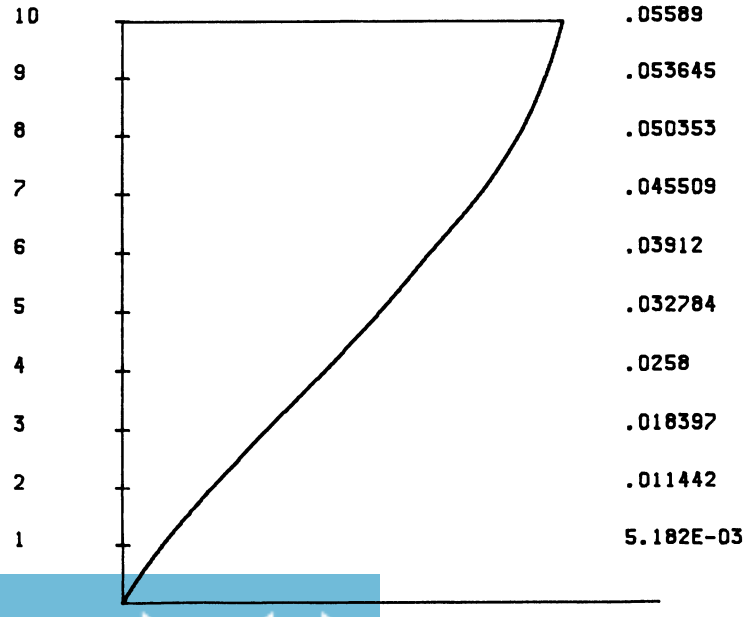
FIGURE 3



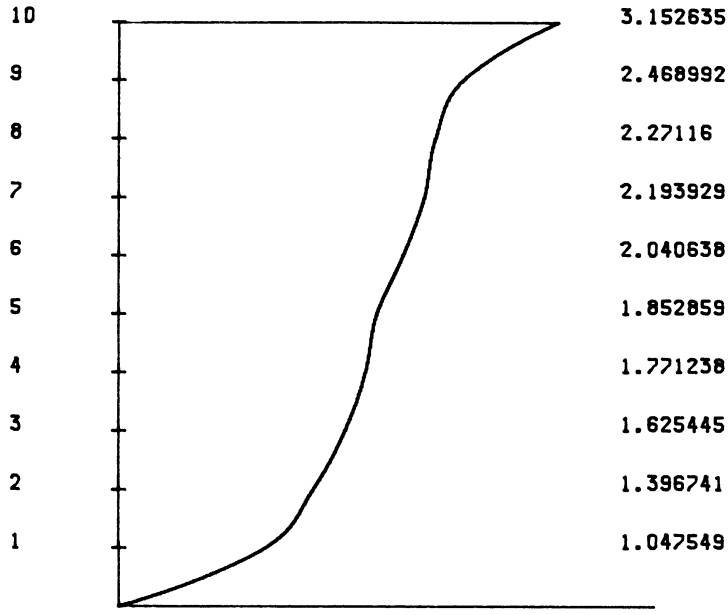
MODE N. 10



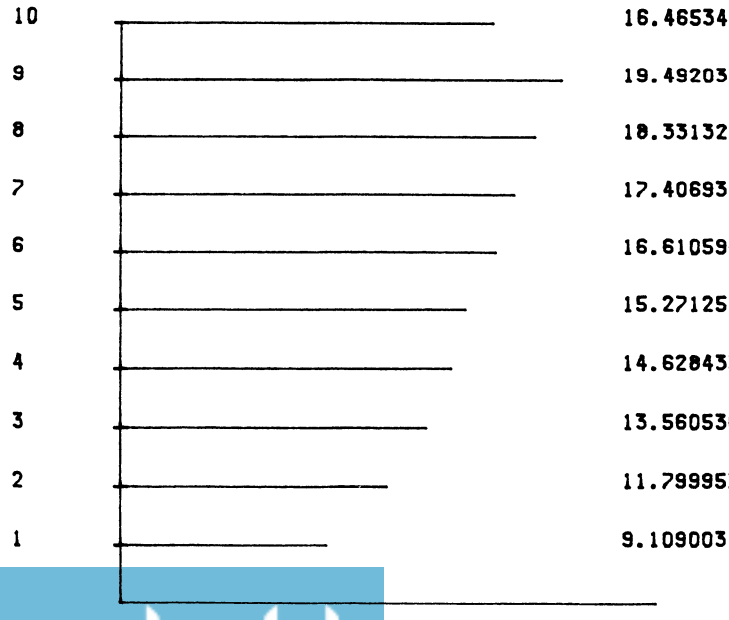
MAXIMUM DISPLACEMENTS



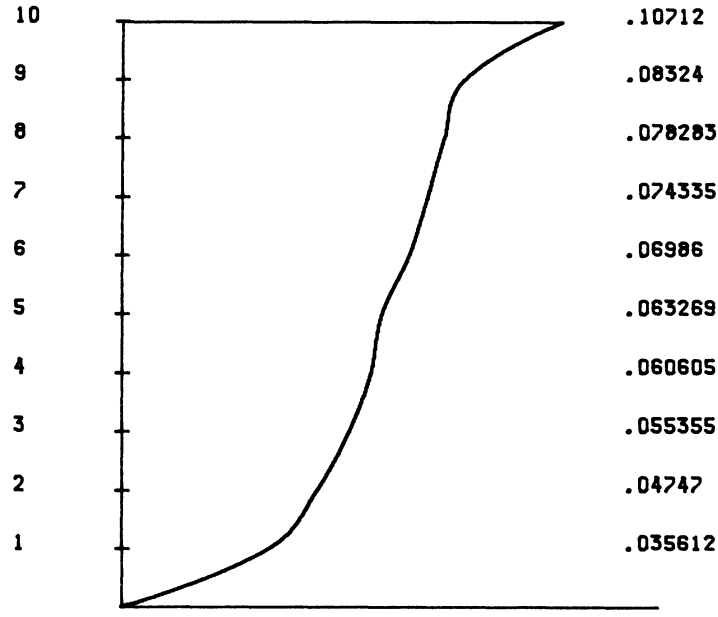
MAXIMUM ACCELERATIONS



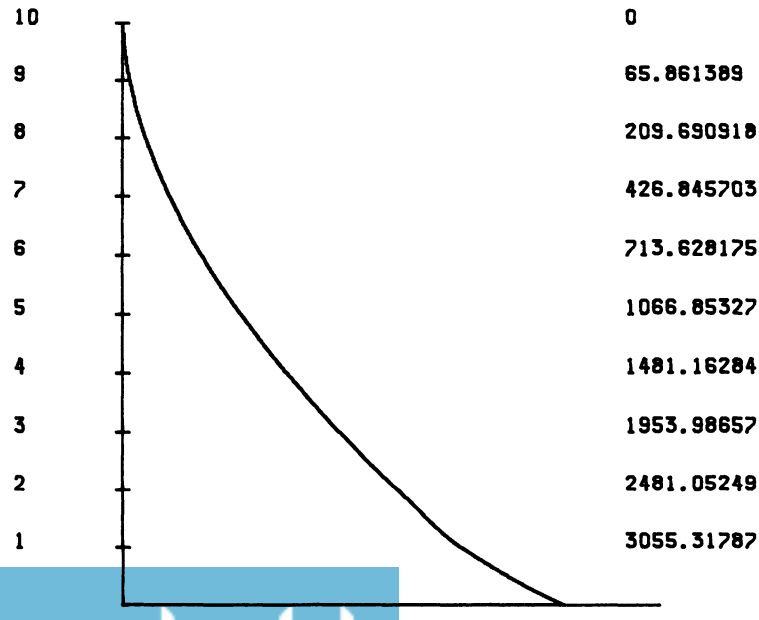
EARTHQUAKE FORCES



SEISMIC COEFFICCIENTS



TOTAL BENDING MOMENTS



VIBRATION ANALYSIS OF ELASTIC ROTATIONAL SHELLS USING MICRO-COMPUTERS.

R. Delpak,+ V. Peshkam.*

+Principal Lecturer in Civil Engineering, *Research Assistant in Civil Engineering, The Polytechnic of Wales, CF37 1DL, Wales, U.K.

ABSTRACT

An element was developed previously which is capable of predicting the undamped natural frequencies of thin elastic shells of revolution. The above element possessed a number of features, e.g. a change in element characteristics could be achieved by a change in the input data. The above formulation has now been updated so that the internal degrees of freedom have been condensed. Also an efficient eigenvalue-economiser routine has been developed to reduce the iteration time. The routine is capable of selecting the masters analytically at a given cut off frequency. These have facilitated the storage and running of the above routines on modern micro-computers. A number of well established examples have been tackled and the results are given in the text.

INTRODUCTION

The extent of interest in subjecting the thin rotational shells to static and dynamic loads has been historical and progressive. This interest is shown by the practising engineers and the engineers/researchers alike. As can be seen, an increasing number of structures of this type are being analysed and constructed in Civil, Mechanical and Chemical Engineering fields. The safety consideration of these structures have involved the well established methods of functional analysis (Reissner 1955, Zarghamee and Robinson 1967), energy methods (Strutt 1933, Arnold and Warburton 1949), the finite difference method (Hashish and Abu-Sitta 1971) and more recently the finite element method (Sen and Gould 1974, Delpak 1980).

The various element formulations for rotational shell analysis using the finite element method is well known and a sample of relevant literature could be found in the enclosed reference. The analysis of realistic shell problems have involved computer usage of considerable extent, both at element development and

at data processing stages. Therefore these engineers who lacked a computer of adequate size, wishing to have access to finite elements results, had to refer to other establishments for their requirements. With recent manufacture and marketing of micro-computers, it is now possible to install a small size processor in any design office. Thus it is essential that the well proven programs should be "stream lined" in order to be put on micro-computers available to consulting engineers.

The present finite element program has been developed so, that frugal use of core memory is maintained at all times. The initial structuring and laying out of various programs and the choice and modification of the processing subroutines, have involved substantial thought and planning.

ELEMENT FORMULATION

The purpose of the present note is not to discuss the element formulation which has been covered elsewhere (Delpak 1975, Love 1944), but is to expand on some other features. However, a fleeting reference might be thought appropriate for inclusion.

The formulation is of isoparametric type, the parent element which has the functional Φ_i and its derivative $\left(\frac{d\Phi}{d\xi}\right)_i$ is normalised at nodes (1) and (2) so that,

$$\begin{cases} \xi = -1 \\ i = 1 \\ \Phi_i = \Phi_2 \\ \left(\frac{d\Phi}{d\xi}\right)_i = \left(\frac{d\Phi}{d\xi}\right)_1 \end{cases} \quad \text{and} \quad \begin{cases} \xi = +1 \\ i = 2 \\ \Phi_i = \Phi_2 \\ \left(\frac{d\Phi}{d\xi}\right)_i = \left(\frac{d\Phi}{d\xi}\right)_2 \end{cases}$$

and elsewhere

$$\phi(\xi) = \sum_{i=1}^{2} \left\{ \phi_i N_i(\xi) + \left(\frac{d\phi}{d\xi}\right)_i N_i'(\xi) \right\} . \qquad 1(a)$$

where $N_i(\xi)$ and $N_i^1(\xi)$ are interpolating functions. These shape functions are third-order Hermitian polynomials which are given as the basic functions $B_1 - B_4$. The parent element and the basic functions are depicted in Figures 1 and 2. The explicit response of the functional Φ everywhere is,

$$\phi(\xi) = \left\{ \phi_1 \cdot \tfrac{1}{4} \cdot (\xi^3 - 2\xi + 2) + \left(\frac{d\phi}{d\xi}\right)_1 \cdot \tfrac{1}{4} \cdot (1-\xi)^2 (1+\xi) + \right.$$

$$\left. \phi_2 \cdot \tfrac{1}{4} \cdot (-\xi^3 + 3\xi + 2) + \left(\frac{d\phi}{d\xi}\right)_2 \cdot \tfrac{1}{4} \cdot (1-\xi)(1+\xi)^2 \right\} . \qquad 1(b)$$

Figure 1

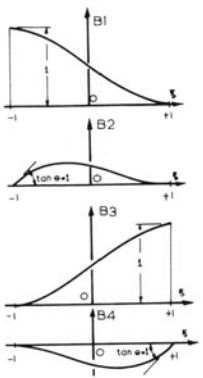


Figure 2

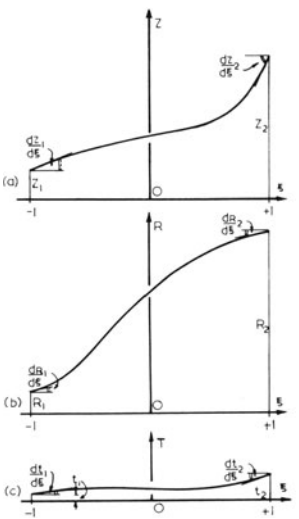


Figure 3

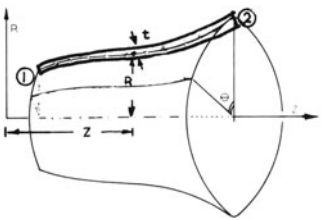


Figure 4

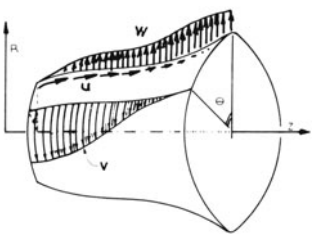


Figure 5

Since the element is isoparametric, the variation of the geometric variables and the displacements also follow the above variation. The geometrical variables Z and R and t are shown in Figure 3(a)-(c) and are

$$Z(\xi)=\sum_{i=1}^{2}\left\{Z_i N_i(\xi)+\left(\frac{dZ}{d\xi}\right)_i N_i'(\xi)\right\}, \qquad 2(a)$$

$$R(\xi)=\sum_{i=1}^{2}\left\{R_i N_i(\xi)+\left(\frac{dR}{d\xi}\right)_i N_i'(\xi)\right\}, \qquad 2(b)$$

$$t(\xi)=\sum_{i=1}^{2}\left\{t_i N_i(\xi)+\left(\frac{dt}{d\xi}\right)_i N_i'(\xi)\right\}. \qquad 2(c)$$

The possible variation of coordinates for a typical element is shown in Figure 4. The variation for displacements in u(along Z), w(along R) and v (along θ) is shown in Figure 5 and given as follows.

$$u(\xi)=\sum_{i=1}^{2}\left\{u_i N_i(\xi)+\left(\frac{du}{d\xi}\right)_i N_i'(\xi)\right\}, \qquad 3(a)$$

$$w(\xi)=\sum_{i=1}^{2}\left\{w_i N_i(\xi)+\left(\frac{dw}{d\xi}\right)_i N_i'(\xi)\right\}, \qquad 3(b)$$

$$v(\xi)=\sum_{i=1}^{2}\left\{v_i N(\xi)+\left(\frac{dv}{d\xi}\right)_i N_i'(\xi)\right\}. \qquad 3(c)$$

In addition to the above points, the element enjoys a set of heirarchical displacements, the inclusion and numbers of which depend on numerical input for every element. These displacements are generated according to engineer's requirement from a set of Legendre-type functions devised by Irons 1965. The actual formulation of these functions, nick-named as surplus functions are presented in Reference (4) but, the visual representation is given in Figure 6. All displacements could be chosen from any of the surplus functions both in local and global coordinates. A typical selection of u, w and v is depicted in Figure 7.

The displacements assumed in the solution of the shell problems are given by two sets of displacements,

$$\{\delta\}_{\theta\text{-sym.}}=\begin{Bmatrix}u.\cos(n\theta)\\ w.\cos(n\theta)\\ v.\sin(n\theta)\end{Bmatrix} e^{i\omega t}, \quad \{\delta\}_{\text{anti-sym.}}=\begin{Bmatrix}u.\sin(n\theta)\\ w.\sin(n\theta)\\ v.\cos(n\theta)\end{Bmatrix} e^{i\omega t} \qquad 4$$

The first set of which can solve axisymmetric and asymmetric problems when $n = o$ and $n \geqslant 1$ respectively. Torsion problems are solved using the second set when $n = o$.

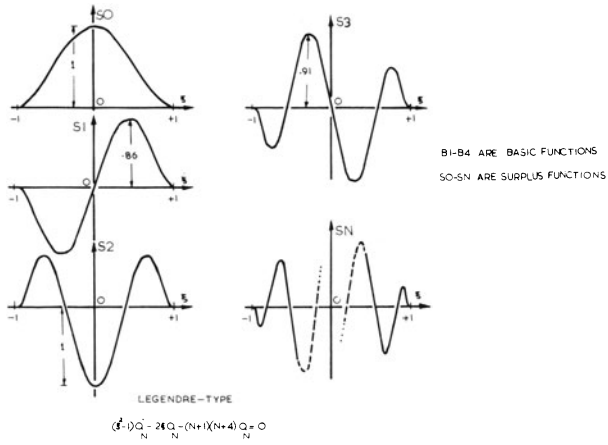


Figure 6

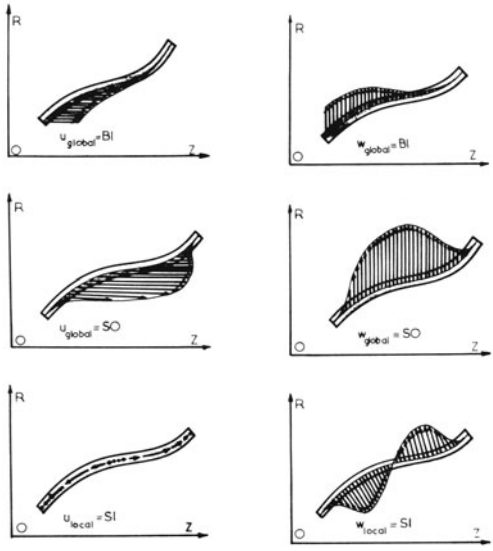


Figure 7

GENERATION OF STIFFNESS AND MASS MATRICES

Owing to variable thickness and other considerations, it was decided to generate the stiffness and the mass matrices using numerical integration namely the Gaussian type. The full account of the above is given in Reference (4). It would suffice to say that the above matrices were generated from the following standard equations

$$[K] = \int_V [B]^T [D] [B] \, d(vol), \qquad 5\ (a)$$

and

$$[M] = \int_V [N] [\rho] [N] \, d(vol). \qquad 5(b)$$

By suitable choice of material property parameters, it is possible to cater for anisothropy, e.g. arthotropically reinforced concrete shells. This will be reflected in the generation of $[D]$, the elastic matrix. The differential operator $[B]$ is composed of linear terms only and are originally from Love, 1944 and Novozhilov, 1964. In generating the mass matrix, the inertial kinetic energy is neglected and only the translational components are included. No adjustment to density matrix is possible except for variations along the generator.

The following is the method by which the Gaussian quadrature is implemented,

$$I = \int_V (\ldots) dv$$

$$= \int_\theta \int_S (\ldots) t R \, ds \, d\theta \quad \text{(ds is an infinitesimal element along generator)}$$

$$= \int_\theta \int_{-1}^{+1} (\ldots) t(\xi) R(\xi) \left(\frac{d\xi}{d\xi}\right) d\xi \, d\theta$$

$$= \int_\theta \left[\sum_{i=1}^{n} (\ldots)_i t_i R_i \left(\frac{ds}{d\xi}\right)_i H_i \right] \cdot d\theta \quad \text{(Page 198 Reference 12)}$$

The integration with respect to θ is simplified in view of the decoupled trigonometric terms so that the standard expressions are obtained as follows,

$$I^1 = \int_\theta (\ldots)d\theta = (\ldots).2\pi \text{ for axisymmetric case}$$

$$I^1 = \int_\theta (\ldots)\text{Cos}^2(n\theta)d\theta = \int_\theta (\ldots)\sin^2(n\theta)d\theta = (\ldots)\pi$$

for asymmetric case

Once $[K]^e$ and $[M]^e$ in Eqn.5 are integrated, the total degrees of freedom will consist of "b" basic functions and "s" surplus functions so that,

$$(\text{d.o.f.})^e = b + s$$

For example in an asymmetric problem with two surplus u's and four surplus w's and v's each, the size of $[K]^e$ and $[M]^e$ would be

$$(\text{d.o.f.})^e = 3 \times 4 + (2+4+4)$$
$$= 22$$

The choice and the numbers of degrees of freedom per element is a matter of input data rather than change in formulation so that the degrees of freedom per element could be varied for different elements within the same problem.

ECONOMISERS

The mathematical modelling of structures to any degree of realism using numerical methods is likely to lead to a large number of degrees of freedom, which might render the micro-computers inefficient. It is therefore essential to adopt methods and equations which are frugal in number of operations involved, or the storage needed and preferably a combination of both. The problem of economising is therefore handled both at the elemental and at the structural levels.

(a) Condensation at the elemental level

The concept of condensation of the internal degrees of freedom in terms of nodal variables is not a novel idea. The implementation is based on formulation such as those given in References (8)(5)(13). Applications have been made for static and vibration problems typified by Popov and Sharifi 1970, Zienkiewicz et al 1967. The present work has involved applying the above technique for both $[K]^e$ and $[M]^e$ so that the following degrees of freedom are eliminated,

(1) $\frac{du}{d\xi}$ for all axisymmetric problems
(2) $\frac{du}{d\xi}$, $\frac{dv}{d\xi}$ for all asymmetric problems
} Using slope functions B2,B4

(3) All d.o.f. resulting from $S_o - S_n$ (Figure 6)

After condensation the total number of degrees of freedom per element given in Equation 6, is reduced to the following sized symmetrical blocks ready for assembly.

$[K]^e_{6x6}$ $[M]^e_{6x6}$ contain u_i, w_i and θ_i (i=1,2) for axisymmetric problem

$[K]^e_{8x8}$ $[M]^e_{8x8}$ contain u_i, w_i, v_i and θ_i (i=1,2) for asymmetric problem

$[K]^e_{2x2}$ $[M]_{2x2}$ contain v_i (i=1,2) for torsion problem.

(b) Elimination of unwanted variables at structural level

The relation governing the elimination of variables at the structural level are fundamentally identical with those at the element level. Except that, the degree of freedom elected for elimination contributes only to those variables which are associated with it. The implementation of this technique has been first carried out by Zienkiewicz et al, 1967.

The most widely used method of reducing the size of the eigenvalue problem, is due Irons, 1963 and Guyan, 1965, which is an approximation preserving the low frequency spectrum. Generally, the selection of masters is left to the expertise and judgement of an engineer. The present routine is capable of selecting the masters analytically, the full account of which is given in Reference(14). However, the option of selecting the masters manually has also been catered for.

EXAMPLES

The number of examples which could demonstrate the relative merits of this type of program is potentially inexhaustable. This could be incorporated by varying (a) the generator shape, (b) the type of loading and (c) specifying whether axisymmetric, asymmetric or torsional. However, six examples were chosen which were thought to represent the overall performance of the program. These are given as follows.

Long cylindrical shell analysed as a beam

The structure in the form of a long tube is tested extensively as (a)S.S. beam,(b) an axtensional rod and (c) a thin walled shaft in torsion. The theoretical reference values are calculated from elementary mechanics. The F.E. results are presented in Table 1. It must be noted that the results obtained are only with few d.o.f., namely 10 or 12 respectively.

Short cylinder fixed round both edges

This example is covered in a variety of publications. The significance of inclusion in the present work is for comparison purposes only. The change in frequencies with the number of nodal diameters is given in Figure 8. The two modes shown by solid lines are obtained using symmetry. If the full cylinder were to be analysed, the true second mode will appear which is unreported by the previous published work. The latter can also be obtained by appropriate imposition of boundary conditions on the half cylinder.

Vibration of cones in flexure

The frequencies of this geometry is determined in previous work (a) experimentally (Weingarten, 1965), (b) mathematically (Strutt, 1933) and (c) numerically (Sen and Gould 1974, Delpak 1980). The present values are determined using only a total of 12 d.o.f. The results and comparisons are given in a graphical form, see Figure 9.

Bending vibrations of thin discs

In order to retain brevity, the pin edge supported disc was analysed. The results are shown in Table 2. The maximum number of d.o.f. were 14. However, severe condensation was imposed both manually and analytically. The cut off frequency was 670 rad./sec.

Fixed spherical cap

The natural frequencies for the spherical caps have been determined for clamped edges conditions owing to relative ease of solution. The results are generally given in terms of base frequency ω_o so that $\omega_o = \frac{1}{a}\left[E/\rho(1-\nu^2)\right]^{1/2}$ (See Kraus 1967, Zarghamee and Robinson 1967). A comparison in graphical form is given in Figure 10. The F.E. results both from previous and present work give consistently different 4th mode frequencies at n = o.

Natural frequencies of the cooling tower

It is rewarding to compare the results from the four previous analyses with the present work as follows (a) numerical integration (Carter et al 1968), (b) finite difference method (Hashish and Abu-Sitta 1971), (c) finite element method (Sen and Gould 1974, Delpak 198)). The results which are obtained using a maximum of 20 d.o.f. are given in Figure 11.

Table 1
Angular Frequencies w rad/sec

Problem Type	Bending			Ext.			Torsion		
Mode No.	1	2	3	1	2	3	1	2	3
Theor.	11.03	44.13	99.31	49.67	149.02	248.36	31.42	92.24	157.07
F.E.	12.11	54.17	138.28	53.40	360.87	580.32	34.64	-	-
d.o.f.	4	4	4	4	4	4	1		
F.E.	10.33	41.80	101.94	50.12	164.43	315.41	31.45	95.13	161.15
d.o.f.	12	12	12	10	10	10	10	10	10

$L = 100$ $E = 10 \times 10^6$
$R = 5.0$ $\nu = 0.25$
$t = 0.1$ $\rho = 1.0$

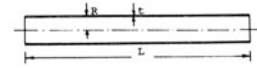


Table 2

Angular Frequencies rad/sec

dia. nodes	n = o			n = 1			n = 2			n = 3		
mode nos. / Prev. wk. / Pres. wk.	m = 1	m = 2	m = 3	m = 1	m = 2	m = 3	m = 1	m = 2	m = 3	m = 1	m = 2	m = 3
	8.358+ 8.400*	+ 49.91	+ 125.52	+ 23.37	+ 81.36	+ 174.40	+ 43.02	+ 117.73	+ 233.54	+ 67.07	+ 159.91	+ 289.66
Majority 5 d.o.f.	8.38	52.43	140.05	23.61	93.98	241.42	44.24	139.30	316.80	71.68	189.14	397.17
Majority 9 d.o.f.	8.36	50.47	129.81	23.47	83.31	201.74	43.30	120.90	262.69	67.82	166.21	323.51
Majority 13 d.o.f.	8.36	50.02	126.19	23.45	81.97	176.42	43.23	119.30	235.96	67.55	162.20	310.27
Majority 8 d.o.f.	8.36 x	x 50.02	x 126.19	x 23.45	x 81.97	x 176.42	x 43.23	x 119.30	x 235.96	x 67.55	x 162.20	x 310.27
Majority 6 d.o.f.	x 8.29	x 63.63	x 182.44	o 23.45	o 81.98	o 176.62	o 43.23	o 119.50	o 236.28	o 67.55	o 162.51	o 312.63

* Determined approximately by Rayleigh - Ritz Method.
\+ Source from Reference 2.
x Manual condensation.
o Analytic condensation, cut-off freq. 670 rad/sec.

Frequencies of a simply supported plate in radious/sec.

$\rho = 1.0$ $E = 30 \times 10^6$ $\nu = 1/3$

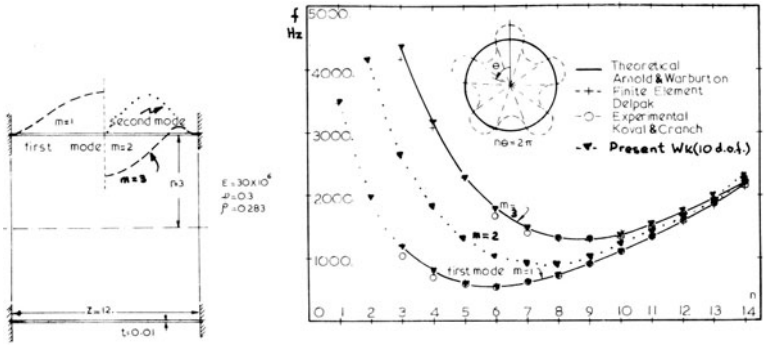


Figure 8

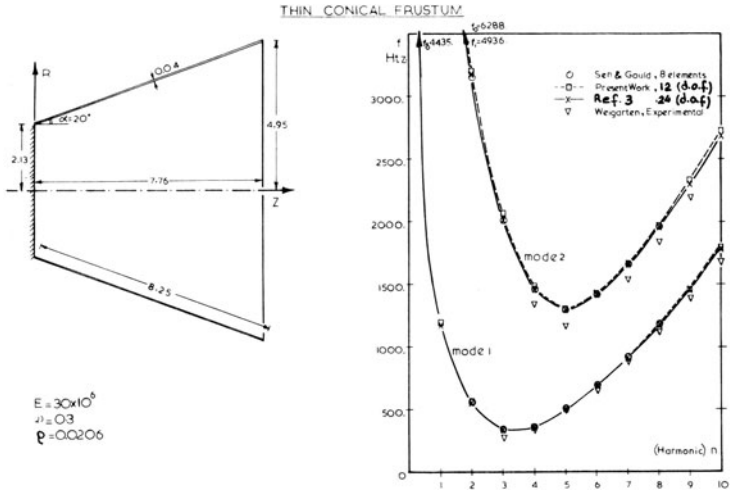


Figure 9

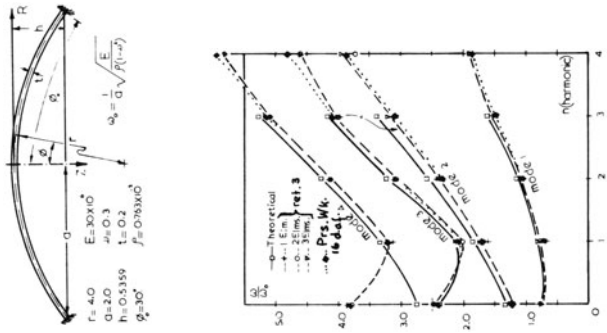


Figure 10

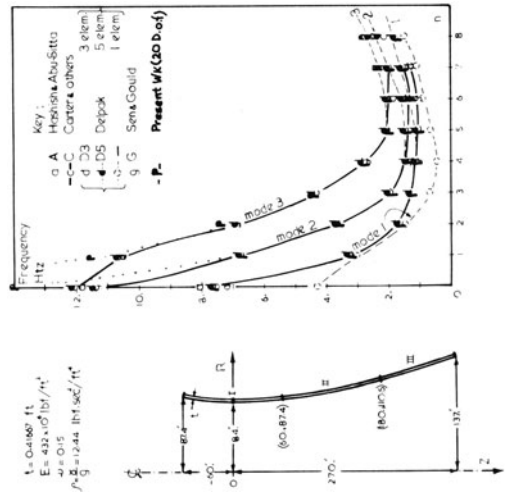


Figure 11

CONCLUSIONS

The most suitable microcomputer for use in the present work was a Sirius 1 using FORTRAN as the source language.

The eigen-value economiser routine (i.e. analytical and manual selection of masters) has proven to be efficient which is an additional source of reducing the iteration time.

It can be seen that the numerical results obtained from the present work, suffer little or no loss of accuracy by incorporating the new features, despite a significant reduction in problem size.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Mr. R.D. McMurray and his Departmental Staff for making the present work possible. They also wish to acknowledge the staff of the Polytechnic Computer Centre for all their assistance.

REFERENCES

1 - Arnold, R.N. and Warburton, G.B. (1949) Proc. Roy. Soc., 197, 238.

2 - Carter, R.L. et al (1968) Free and Forced vibrations of hyperbolical shells of revolution.Struc.Res.Ser., No. 334, Civil Eng. Studies, Univ. of Illinois.

3 - Delpak, R. (1980) A finite element assessment of natural frequencies of undamped elastic rotational shells. App. Math. Mod., 4, 357-368.

4 - Delpak, R. (1975) Ph.D. Thesis CNAA, Lond.

5 - Guyan, R.J. (1965) Reduction of stiffness and mass matrices AIAA J., 3, 3:380.

6 - Hashish, R.N. and Abu-Sitta, S.H.J. (1971) J.Eng.Mech.Div. ASCE, 97, 253.

7 - Irons, B.M. (1965) Shape functions for elements with point conformity. A.S.M., 1204, Rolls-Royce Library.

8 - Irons, B.M. (1963) Eigenvalue economisers in vibration problems. J.Roy. Aer. Soy., 67, 526-528.

9 - Kraus, H. (1967) Thin elastic shells. John Wiley and inc., New York, 237-331.

10 - Love, A.E.H. (1944) A treatise on the mathematical theory of elasticity. Dover, New York.

11 - Novozhilov, V.V. (1964) Thin shell theory. Noordhoff, The Netherlands.

12 - Popov, E.P. and Sharifi, P. (1970) A refined curved element for thin shells of revolution. John Wiley & Sons, Ltd.

13 - Przemieniecki, J.S. (1968) Theory of matrix structural analysis. McGraw Hill, New York, 147-148.

14 - Reissner, E. (1955) Quarterly of App. Math., 13, 279

15 - Sen, S.K. and Gould, P.L.J. (1974) Eng. Mech. Div. ASCE, 100, 283.

16 - Shah, V.N. and Raymund, M. (1982) Analytical selection of masters for the reduced eigenvalue problem. Int. J. Num. Meth. Eng., 18, 89-98

17 - Strutt, M.J.O. (1933) Eigenschwingungen der Kegelschale. Annalen der Physik, 17, 729-735.

18 - Weingarten, V.I.J. (1965) Eng. Mech. Div. ASCE, 91, 69.

19 - Zarghamme, M.S. and Robinson, A.A. (1967) A numerical method for analysis of free vibration of spherical shells. AJAA J., 5, 7:1256-1261.

20 - Zienkiewicz, O.C. (1977) The finite element method. McGraw Hill, Lond. & New York.

21 - Zienkiewicz, O.C., Irons, B.M. and Anderson, R.G. (1967) Vibration and stability of plates. Research report No. C/R/78/67 Univ. of Wales, Swansea.

COMPUTER AIDED STATIC AND FATIGUE FULL-SCALE TESTS: POSSIBLE APPLICATIONS OF DESKTOP COMPUTERS

A. Nappi

Department of Structural Engineering,
Technical University (Politecnico), Milan, Italy

ABSTRACT

The paper deals with the extensive use of a desktop computer in a recently developed testing facility. The present configuration has been designed for static tests of airframes and for fatigue tests of aircraft structural components.

Only one desktop computer has been installed and a rather sophisticated program allows the calculator to perform all the operations needed to carry out a a structural test (load control, data acquisition, output of significant results). Hydraulic actuators are used to reproduce the load conditions selected for testing structures; the control is ensured by the computer through a feedback system.

The main features of the computer code are discussed and particular emphasis is given to the possibility of performing data acquisition and load control at the same time with a single desktop computer. Also the electronic instruments chosen for the test rig are briefly described. Details are also given about those performances of the testing equipment which depend on the present version of the computer package.

INTRODUCTION

The testing facility described in this paper was conceived† during the development program of a new jet trainer (S.211), which made its first flight on April 10th, 1981.

During the preliminary studies of the testing pro-

---

† At that time (April 1980 - September 1981) the Author was in charge of the Ground Testing Department of SIAI Marchetti, Sesto Calende, Italy.

gram, priority was given to static tests; it soon became evident that many advantages would be provided by a fully automatic feedback control system for the applied loads, as happens in modern fatigue testing facilities. A desktop computer was selected, since the speed and the computational effort required for the application of quasi-static loads is relatively small. It was also decided that hydraulic actuators, load cells and servovalves suitable for fatigue testing had to be chosen. Thus, it could be easier to develop the equipment so that, in the future, it would also be possible to perform fatigue tests, without changing all the components of the hydraulic system. Furthermore, the same desktop computer selected for static tests could be used for simple fatigue tests requiring only few actuators (as for example happens with structural components, since they do not need so many hydraulic jacks as complete airframes).

The problems concerning data logging were also studied with care because of their importance in the design of testing facilities. A flexible hardware configuration was chosen, so that the system could be easily developed in the future. For the initial configuration a target of 360 measurements was set. In order to cover a wide range of data acquisition speeds and accuracy, two digital voltmeters were selected, one of which could take up to 1000 readings per second. Clearly an efficient controller was needed; however, when the capabilities of modern desktop computers were examined, just one calculator seemed to be sufficient for simultaneous load control and data acquisition (for details see the Appendix).

Software problems were made easier by the choice of a computer which could interrupt (automatically, at given intervals) the execution of a program and activate another routine. In this way, during static or fatigue tests, it would be fairly easy to alternate load control algorithms with the measurement of stresses and/or displacements.

Finally, the same computer could be used also for the presentation of significant results.

Hence, a fully automatic testing facility was designed and the following tasks were established for the master unit (i.e., the desktop computer):

a. Controlling up to 15÷20 hydraulic actuators for static tests and up to 4÷5 cylinders for fatigue tests with cyclic block load sequences (each block being made up of loads with constant amplitudes and mean values)

b. Controlling a multiplexing scanner with the possibility of measuring up to 360 electrical signals by means of a fast or a high resolution digital volt-

meter.

c. Processing the data in order to provide all the significant results.

So far, as it will be pointed out in the following sections, good results have been obtained. In addition, minor changes can permit full-scale tests on complete structures. In fact, fully automatic feedback control techniques have been used and the components of the hydraulic system are suitable for dynamic loading conditions. Of course, a faster computer would be needed and, probably, a more powerful hydraulic pump should be installed.

Therefore, the test rig described here represents an intermediate level between standard facilities for static tests and modern equipments for fatigue tests of complete airframes.

During the development of the test rig an extensive research activity has been carried out in order to investigate the behaviour of the digital control system used for the hydraulic jacks. Load control techniques fully based on digital computers are rather infrequent, but can be an efficient alternative to more usual analog control procedures. However, the most significant feature of the equipment discussed here is not the control system *per se*, but the use of a simple desktop computer for simultaneous data acquisition and load control with several hydraulic actuators. In fact, a fairly new application of desktop computers seems to be suggested here, since, to the Author's knowledge, such computers have not yet been used so extensively in testing facilities. Therefore, the results presented here can offer some rather interesting information about the potential benefits and limitations of desktop computers in this particular type of process control. In addition, the fully automatic equipment for static tests envisaged in this paper may represent an efficient, low-cost solution with some innovating aspects. It should be noted, however, that the basic ideas have been suggested by well-known and well-established techniques currently used for complex fatigue testing on airframes.

LOAD CONTROL FOR STATIC TESTING

When a structure must be subjected to static tests, it is necessary to sort out the most critical load conditions and to reproduce them as accurately as possible. This job can be carried out quite easily by means of computer codes, as shown in Fig. 1, where real, effective distributions of shear and bending moment are compared with those generated through a computer package (Nappi *et al.*, 1980). Shear (S) and

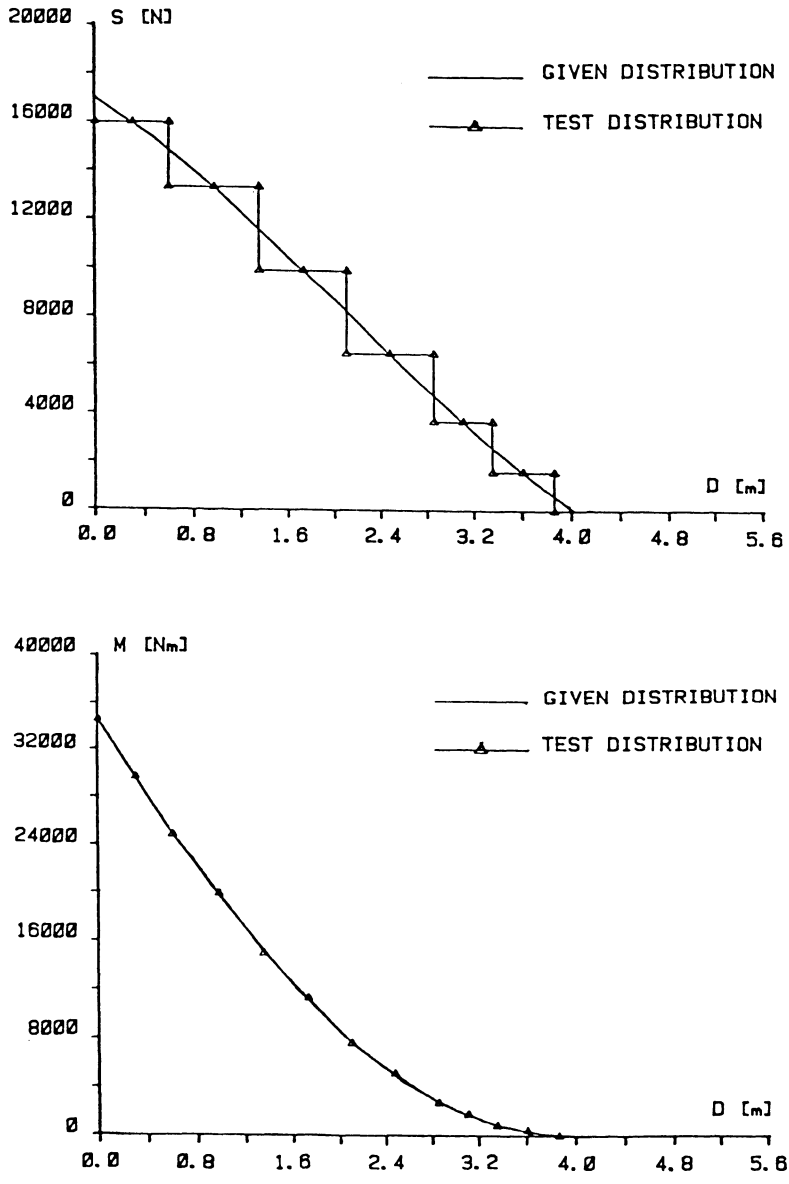


Fig. 1 - Given distributions of shear and bending moment compared with distributions due to discrete forces

bending moment (M) are plotted as functions of a distance D measured along the semispan of a wing.

When testing load conditions have been established, forces are usually applied by means of hydraulic jacks and can be partially distributed through systems of beams, which form a typical "wiffle-tree" arrangement, as discussed, for instance, by Vann (1973). If two or more hydraulic actuators are used, loads must be increased proportionally, to avoid reactions which might induce severe stresses in the structure under test. Hence, it is of paramount importance to obtain a load control system which guarantees not only correct forces, but also simultaneous and proportional load increments (within acceptable tolerances).

In the testing facility described in this paper a feedback control has been designed. The intelligent unit is a desktop computer. It can send input currents to the servovalves (-15÷+15mA) by means of digital to analog (D/A) converters. These guarantee an accuracy of ±10μA and act as programmable current sources. In fact, after receiving a command from the computer, they keep on generating the programmed currents until a different instruction is not given (it is no matter what the calculator is doing in the meantime).

The computer also controls the applied loads by means of analog to digital (A/D) converters. These are connected to the load cells and receive signals ranging from -10 to +10V, with a resolution of 5mV.

For the load control a great effort has been made with a view of realizing a general purpose algorithm, suitable for all the actuators used in the testing facility. This goal has been achieved by developing a feedback control technique based on very small corrections of each input current, so that moderate changes of the hydraulic flow rates can be imposed. Thus, the desired load levels are reached through a sequence of small corrections of the input currents. Each control loop is activated at intervals of 0.5÷1s and requires about 25ms for each actuator. Experimental tests seem to be very encouraging, since tight tolerances are usually feasible. For instance, when loads close to 100KN are applied, tolerances of ±300N can be ensured.

A load level is generally expressed as a percentage of limit loads. When the percentage must be changed, the computer gradually updates the load intensity which is used as reference level (or "target level") by the control algorithm. Therefore, hydraulic flow rates are constantly varied in order to obtain the intermediate reference levels, which are changed proportionally (with small increments) as soon as all the forces are within imposed (rather tight) bounds.

At the end of the process, the desired load level is reached with proportional variations of the forces (at least within given tolerances), so that undue stresses in the structure under test are avoided.

This method of changing the load levels is susceptible of a rather important improvement, which is now being developed by the staff of the testing establishment. The improvement consists in obtaining certain load variations in given times. The algorithm which performs this task is still being tested, but encouraging results have been obtained, as shown in figures 2 and 3. Fig. 2 compares an ideal, perfectly linear ramp with the measured one. The errors with respect to the ideal ramp are shown in Fig. 3 on a different scale. Clearly, they represent a small percentage of the applied loads. In figures 2 and 3 no measured loads appear in the last part of the graph; in fact, measurements were taken only until the load fell within a given tolerance (12000±150N). Therefore some measures were missed during the last few seconds of the ideal ramp, since the actual load was within the desired range.

At significant load levels the response of the structure can be measured by means of strain-gage and potentiometer circuits (for details see, e.g., Dally and Riley, 1978).

As pointed out before, data logging and load control must be carried out simultaneously. This is probably the most difficult task of the desktop computer and has been made easier by the excellent capabilities of the calculator selected for the testing facility. In fact, it allows to interrupt the execution of a program and to activate a different routine at given time intervals. Thus, when a significant load level is reached, the computer performs the data acquisition, stores the measured values on a flexible disk and plots some significant stresses and/or displacements. In the meantime, at given intervals (again 0.5÷1s) the load control procedure is activated and hydraulic flow rates are immediately changed if something is going wrong. Given tolerances are checked carefully. If a load falls out of the prescribed range during data acquisition, this operation is stopped and started again only when all the loads are once more within the desired limits.

The load control procedure discussed here has been conceived, mainly, for static tests of aircraft structures subjected to self-equilibrated loads. During these tests an important source of information is represented by the reactions at the constraints, which must be statically determinate. Therefore, if the loading process is not correct because forces do not change proportionally, reactions tend to increa-

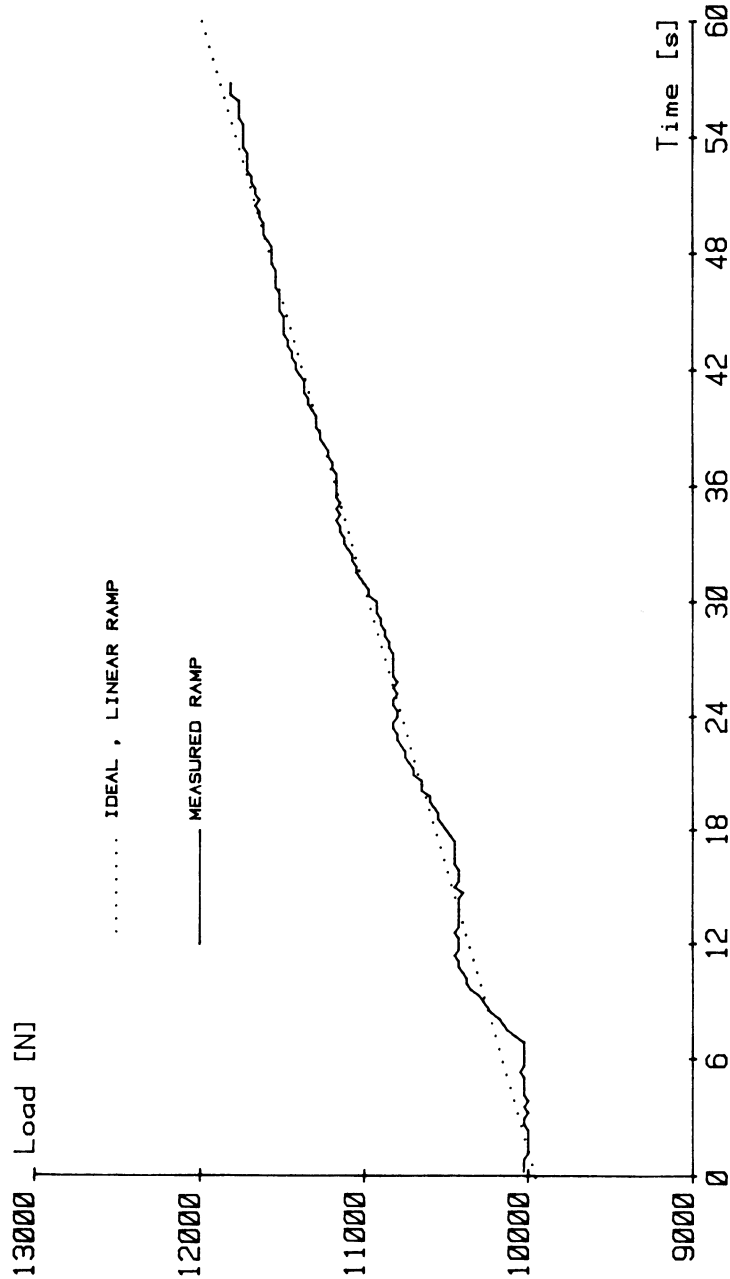


Fig. 2 - Comparison between an ideal and a measured ramp

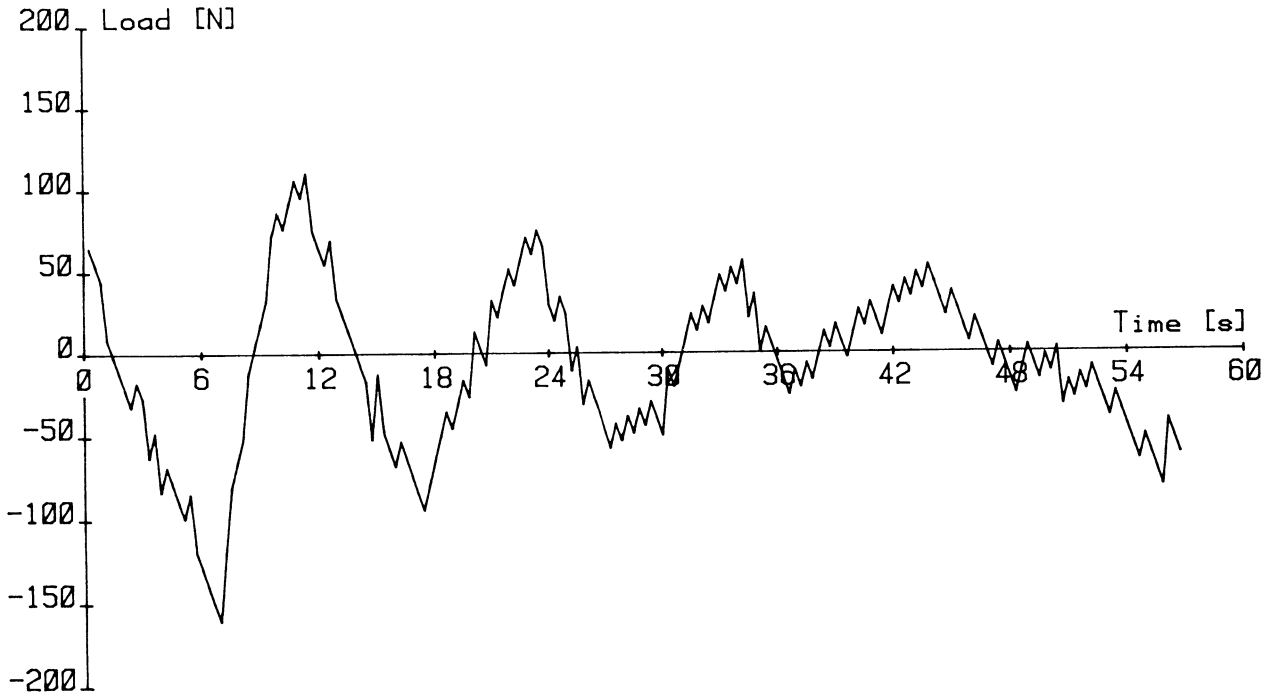


Fig. 3 - Difference between measured and ideal loads during a ramp

se. Hence, in order to avoid any damage caused by undue loads, the computer checks the reactions and automatically stops the test, should they become too different from zero. In fact this is the ideal, theoretical value of each reaction, owing to the self-equilibrated load condition and to the statically determinate constraints.

DEVELOPMENT OF THE LOAD-CONTROL SYSTEM FOR FATIGUE TESTING

As pointed out before, the initial configuration of the testing facility has been designed also for simple fatigue tests. More specifically, load-control algorithms have been studied for few actuators (4÷5) and only constant amplitude block load sequences have been considered.

In the case of fatigue testing, an efficient load control system must perform the following tasks:

a. Realizing minimum and maximum loads within acceptable tolerances

b. Generating in-phase load cycles at given frequencies.

For these purposes the multiprogramming capability of the desktop computer has been exploited once more, to ensure that frequency be exactly as required by the operator. In fact, at given intervals (50÷200ms) different currents are sent to each servovalve, in such a way that a step sinusoidal input is generated. For instance, if the period (t) of a cycle is divided into time intervals τ, at the beginning of each interval the following input current is assigned:

$$i_j = i_{m,j} + \frac{i_{a,j}}{2} \sin \frac{2n\pi\tau}{t} \qquad (1)$$

where $i_{m,j}$ is the mean value of the input current and $i_{a,j}$ its amplitude, while $n=1,2,\ldots$ and $j=1,\ldots,s$ if s represents the number of servovalves.

Every input current is applied during a whole interval τ and is changed at the beginning of the subsequent interval. It is intuitive that an almost sinusoidal input current will determine a time-dependent load with a similar behaviour. This is shown in Fig. 4, which concerns the initial 15 cycles of a test carried out with a single hydraulic jack at a frequency of 0.4Hz. This value is quite low, but is typical of fatigue tests on airframes, as reported, for instance, by Ketola (1965). In Fig. 4 dots represent a nondimensional input current i^* and the full line refers to a nondimensional load P^*. Nondimensional quantities are defined according to the rela-

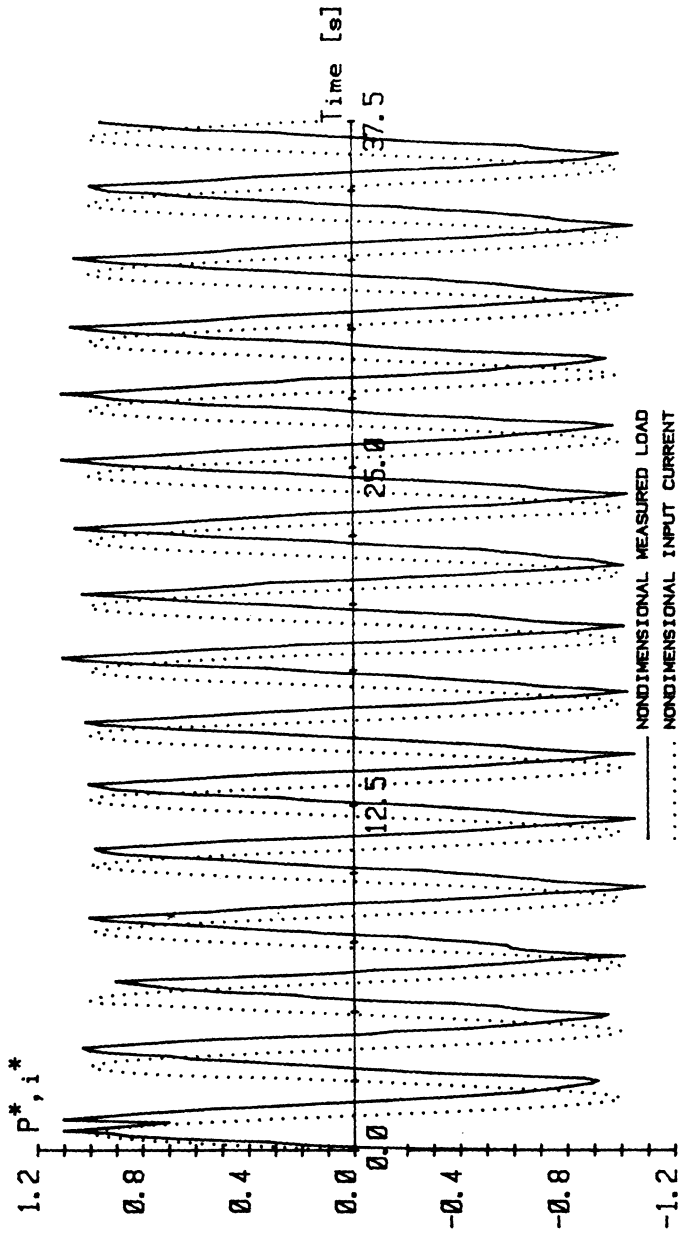


Fig. 4 - Nondimensional input currents and measured loads during the initial 15 cycles of a test

tions:

$$i^* = -2\frac{i - i_m}{i_a} \tag{2}$$

$$P^* = 2\frac{P - P_m}{P_a} \tag{3}$$

In Eq. (3) P is the effective measured load, while P_m and P_a are the mean value and the amplitude of the alternating load which must be realized. Subscripts j's have been dropped since the graph refers to a single actuator. In order to interpret Fig. 4 properly, it should be noted that during the test wire connections were so arranged that decreasing input currents tended to increase tension loads and viceversa.

The computer code activates (at constant time intervals τ) the set of instructions required to apply input currents. Their transmission takes a few milliseconds. The remaining part of each time interval τ can be used for readings of strain gages and potentiometers or for measurements of the applied loads. Such measurements allow to correct the mean values and/or the amplitudes of the input currents, if the response is not as expected. Measured loads also allow to determine the time interval during which the maximum force is attained. Hence, the input current can be shifted along the time axis, in order to reach maximum loads exactly when they are desired. Thus, in-phase responses of the hydraulic jacks can be obtained and no improper load conditions are generated. The efficiency of the technique used by the computer package is proven by the plots of Fig. 4. The purpose was to reach maximum loads at the beginning of each cycle, i.e., at time intervals rt, with r=1,2,... and t=2.5s. Fig. 4 shows that peak values rapidly tend to be shifted towards the correct position, which is reached after ten cycles (or 25s) from the beginning of the test. Thus, the actual input currents will not be given by Eq. (1), but will be defined according to the following formula:

$$i_j = i_{m,j} + \frac{i_{a,j}}{2} \sin \frac{2(n-\nu)\pi\tau}{t} \tag{4}$$

where ν is an integer which varies between 0 and N, if the period t is divided into N time intervals.

Only few cycles are required for the adjustment of maximum and minimum loads. Their precision is guaranteed by correcting the basic input currents as soon as measurements show that loads are increasing or decreasing too much. Fig. 4 does not show any correction of the basic input current, but indicates only

the original sinusoidal shape. However, the actual input current does not always follow the step-sine trend, but is corrected immediately as soon as the load exceeds given thresholds.

It is rather interesting to observe that Fig. 4 refers to a test for which P_m=8000N and P_a=4000N. Thus, errors in the range ±0.1P* (which occur in the plot of Fig. 4) correspond to ±200N, i.e. to a small percentage of the applied loads.

DESCRIPTION OF THE COMPUTER PACKAGE FOR STATIC TESTS

The computer code has been studied with the aim of making the user's job as easy as possible. This goal has been achieved by preparing an interactive program which asks for simple instructions, gives hints to the operator and displays messages when the user's commands are not correct.

Basically, the computer package is made up of a master program (or higher-level module), which asks for key-words in order to activate one lower-level module or to select a series of lower-level modules for in-line inclusion (when several functions must be executed to perform the task required by the operator). All the modules concerning static tests belong to a single package. They allow the computer to perform data processing, load control and output of significant results. About 4000 statements have been written and an extensive use of the overlay technique has been made, since the whole program cannot be stored in the read-write memory provided by the computer. Therefore, several modules are stored on a flexible disk and are automatically loaded when are needed.

In order to process data as fast as possible, a large amount of measurements can be stored directly in the read-write memory, so that frequent data transfer from mass memory is avoided. About 5000 data are readily available and they include up to ten readings per channel, their mean values, their standard deviations and reference measurements (i.e., the input signals recorded at zero-load level).

Much attention has been given to the input of large sets of data (which are required, e.g., for strain-gages and potentiometers). For these cases special editing procedures have been conceived to allow an easy entry of data. For instance, when strain-gages are defined, the input data (which must include an alphanumerical label to identify each strain-gage) are checked. If no errors are detected, they are stored automatically according to an alphanumerical order (based on the labels). Later on, data can be cancelled or corrected or updated by using simple key instructions and the program automatically arranges the

whole set of data (as happens when program statements are written or revised by means of usual editing procedures).

Some standard sequences of functions have been included. An example is provided by the following operations, which are typical of static tests: load control, data acquisition at significant load levels, data storage and plots of meaningful stresses or displacements. These operations are executed automatically without continuous instructions of the user. In fact, the operator must give simple initial commands only (with the help of interactive preprocessors which ask for the necessary instructions). Then, as soon as the standard procedure is activated, only load levels must be defined (by entering the percentage with respect to limit loads). Data concerning a certain test are stored on files which are easily recognized by means of a code (defined by the user) and of a progressive number (starting from zero). Data files are recorded on flexible disks, which represent very compact media to preserve all the measurements taken during tests.

A standard procedure has been conceived also for the output of results. It can produce a complete list of stresses and displacements concerning a test. The output sequence is fully automatic. It is activated as soon as the user has given some initial instructions and has entered the code which allows the computer to identify the names of the files where the relevant data are stored.

A second program is being developed for fatigue testing, but details are unnecessary here, since its features are quite similar to the characteristics of the program illustrated above.

CONCLUSIONS

The research activity carried out for the development of the computer package described in the previous section and the high level of reliability exhibited during the preliminary tests on the S.211 airframe seem to show that small calculators can also be very efficient for rather complex process controls, as needed in modern testing facilities. Obviously, accurate control cannot be expected for random load sequences and/or many hydraulic actuators, since the speed of small computers is relatively moderate. Nevertheless, the computational burden required for the applications described in the present paper seem to be quite heavy and a particularly severe task is represented by simultaneous load control and data logging. Also in this case, however, the performance of the computer has been excellent.

Automatic control procedures allow a single operator to carry out a test and to produce all the relevant results. In addition, no preliminary adjustments of the servovalves are needed to predict their behaviour, since optimal input currents are selected automatically. Also the cost has been quite reasonable, since one desktop computer can carry out all the functions required during a test and only standard, easily available components have been used.

The design of the testing facility discussed in the paper has been suggested by modern fatigue testing equipment, which relies upon fairly large process computers and fully automatic control techniques. Digital computers can be coupled with microprocessors and/or analog feedback control systems, but such solutions have not been adopted for the equipment described above. On the contrary, all possible efforts have been made in order to realize a compact testing facility with a single control unit. As pointed out before, this equipment represents an intermediate stage between the usual facilities for static tests and advanced rigs for fatigue tests on airframes.

On the basis of the experience gathered till now, the testing facility fully complies with the initial specifications. The excellent performance has been confirmed by full-scale static tests already performed on the S.211 airframe and by a fairly large number of experimental investigations on the algorithm developed for fatigue testing.

ACKNOWLEDGEMENTS

All the research work concerning the testing facility described in the paper has been carried out at SIAI Marchetti, Sesto Calende, Italy. The Author wishes to thank Dr. A. Brena, Dr. A. Ferretto and Dr. A. Perazzo for their permission of publishing the present paper. Also the assistance of Mr. F. Crenna is gratefully acknowledged for his contribution to the development of the computer package.

REFERENCES

Dally, J.W., Riley, W.F. (1978) Experimental Stress Analysis, McGraw-Hill Kogakusha, Ltd., Tokio

Ketola, R.N. (1965) A Full-Scale Structural Fatigue Test Program, J. Aircraft, 2, 5: 407

Nappi, A., Crenna, F., Simone, P. (1980) LC: A Fortran IV Program for Simulation of Given Load Conditions, SIAI Marchetti Scientific Computer Library

Vann, F.W. (1973) A300B Static and Fatigue Tests, Aircraft Engineering, 45, 12: 6

APPENDIX

The system is controlled by a desktop computer which uses a modified BASIC language and offers about 60K bytes of read-write memory.

The use of small computers for data logging is quite frequent. Computer aided systems for data acquisition are currently available. The application envisaged in the paper is based on a simple development of standard data logging equipment. Only D/A and A/D converters have been added, so that load control has been ensured without any relevant cost increase. Of course, the same development could be applicable also to different types of process control. An example may be provided by a structural test involving thermal variations. Clearly, it is always necessary to take the moderate power of desktop computers into account, as shown in the case of load control. In fact, only quasi-static loads and low-frequency alternating loads have been considered.

Fig. 5 gives some details on the configuration of the equipment for simultaneous load control and data acquisition in the case of static tests. These operations are carried out by two modules, which are executed alternatively. Their flow-charts are shown in Fig. 5. Routine 'A' acts as a main program. At given intervals (0.5÷1s), routine 'B' is activated. This represents the load control algorithm: it measures all the forces exerted by the hydraulic jacks and, if necessary, updates input currents. At the end of the control loop, the execution of routine 'A' goes on again, starting exactly where it was interrupted. Most of the times, routine 'A' performs only a dummy loop, during which no significant operation is carried out. However, when data acquisition is required and the loads are within a given tolerance, routine 'A' activates the scanner, stores all the measured data and plots some significant stresses and/or displacements. As pointed out in the flow-chart, data acquisition is stopped if some load falls out of the prescribed limits; in this case, a dummy loop is executed once more until all the loads are again within the given tolerance.

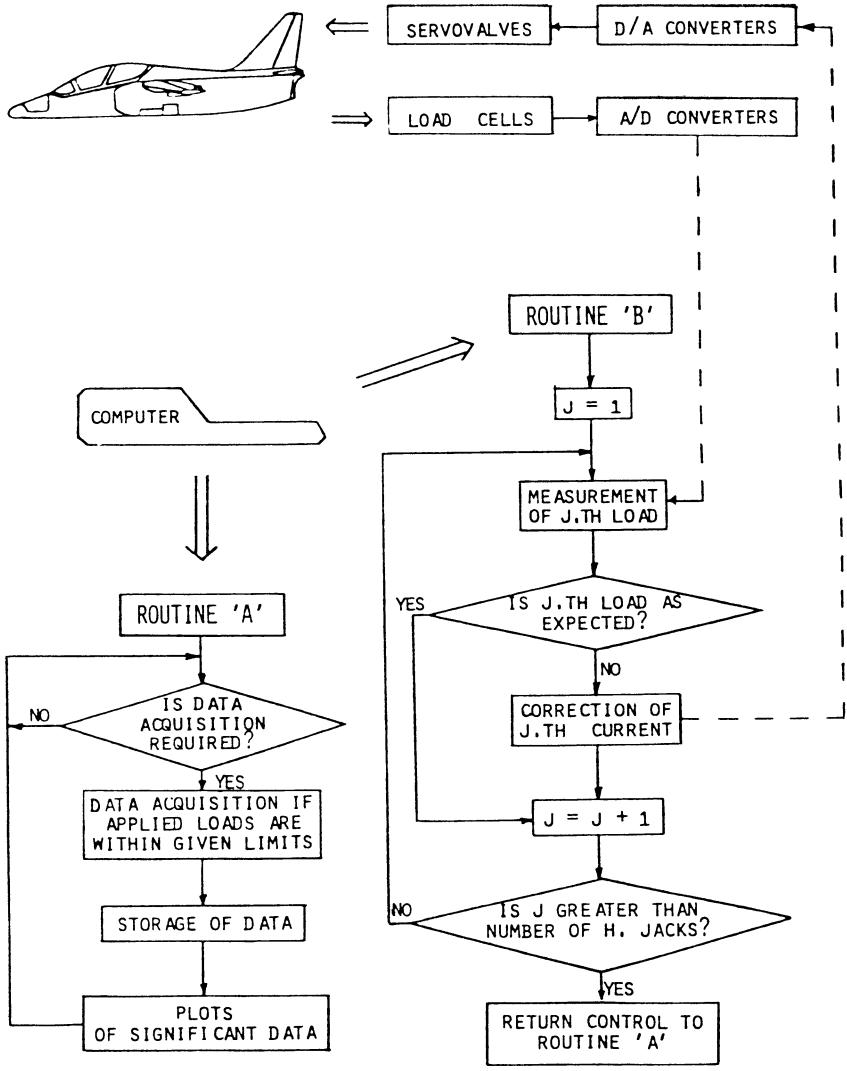


Fig. 5 - Flow-charts of the algorithms for data acquisition and load control

VPCREEP - FINITE ELEMENT ANALYSIS COMPUTER CODE FOR CONCRETE STRUCTURES SUBJECTED TO THE COMBINED INFLUENCE OF CREEP, SUSTAINED AND TIME-VARYING TEMPERATURES

Y.F. Cheng, G.L. England, K.R.F. Andrews

King's College London, U.K.

INTRODUCTION

Where a structure is designed for service at temperatures where creep effects are negligible, a rigorous assessment can be made of the factors which affect integrity, safety and satisfactory performance. Many of the more advanced design codes, such as B.S. 5400, B.S. 3915 and Section III of the ASME pressure vessel and boiler code, provide a sound basis for doing this. The necessary stress analysis methods are available, and the basic properties of the commonly used materials are well established, under both static and dynamic loading. When we come to structures operating at time-varying and non-uniform temperatures, this is no longer true.

Until recently we have known very little about the factors which affect the time-varying and non-uniform temperature performance of structures of complex shape. Design has largely been dictated by experience and by the knowledge that certain components gave satisfactory service. Such experience cannot, by its nature, be extrapolated with certainty to the structures of different shape, or of different size, or in a different material, or subjected to a different operating regime. Any significant advance in plant requirement is therefore accompanied by the risk of either poor structural or uneconomic design or both. The assurances of structural integrity and serviceability are not forthcoming. Fortunately, there has been a growing interest, at least at a research level, in the investigating of structural performance under multi-axial stress states and non-isothermal conditions in the past decade. The combined influence of creep and temperature on the behaviour

of concrete structures has been studied and the importance of these parameters has been demonstrated in experiments relating to statically determinate structures (Suan 1964) for which displacements have been monitored, statically indeterminate beam (Ross et al 1964 and England et al 1977) and portal frame structures (Krishnamoorthy 1971) for which external time-varying redundancies and internal stress redundancies have been recorded. It has been shown that non-uniform temperatures cause stress redistribution with time, this is true even when temperatures and loads are time-invariant.

Furthermore, the powerful mathematical modelling techniques, notably the finite element method, together with those modern super-computers, such as Cray and Cyber 205, have provided the engineer with computing power undreamt of by his predecessors. The engineer is now able to extract precise information on the stress and strain history in the structure and thus provide a rational basis for design.

THE VPCREEP COMPUTER CODE

Historical time-step numerical procedures (England 1967) are capable of predicting the variation of stresses and strains with time, but require many time steps if accuracy of solution is to be maintained throughout the life-span of the structure. In consequence, this type of procedure is often expensive. Published works (Dhalla et al 1975 and Barsoum et al 1976) indicated that the human resources and level of engineering expertise required to perform such an analysis would virtually prohibit the use of three dimensional continuum modelling for routine design and analysis calculations.

Direct solutions (England 1968) are scarce but do exist for some classes of problem. Problems in this class include cases of sustained heating and cooling, and temperatures which change in a step-wise manner between defineable limits and in a cyclic manner. It is assumed for this latter type that temperature transients are of short duration compared to the 'hold' periods at steady-state temperatures between successive changes.

The VPCREEP computer code, or Virtual Power CREEP analysis system, is designed as a special purpose finite element program for the analysis of two and three dimensional concrete structures subjected to the combined influence of creep, sustained and time-varying temperatures utilising the Virtual Power Minimisation procedure (England 1968). This procedure has the advangtage that it may be used in the form of exact or approximate analysis and more economic than the

time-step techniques. It is also possible to strike a balance between cost and precision of solution and thus use the code for both design work and analysis.

THEORETICAL BASIS

Strain Representation for Concrete - For situations in which shrinkage is insignificant, the strains in concrete may be represented as the sum of two components: (i) elastic strain; (ii) creep strain. These strains are influenced by such factors as concrete mix proportions and age, and by environmental temperature. Creep is increased markedly by elevated temperatures (England 1966), and may be treated in analysis, after normalisation with respect to stress (Ross 1958) and temperature (England 1968) on a pseudo-time basis. Pseudo-time, c, is taken to be the normalised creep parameter itself and replaces real time in the analysis. This representation allows the normal ageing creep characteristics with respect to real time to be removed from the analysis and thus reduces the complexity of the creep solution. Conversion to real time is then made at the end of the analysis by reference to the normalised creep/real time data for the concrete.

For this simplified representation the creep behaviour of concrete becomes analogous to that of a non-homogeneous Maxwell material for which pseudo-time c replaces real time and the ageing dashpot viscosity is replaced by the reciprocal of the creep-temperature normalising function $\phi(T)$. Experiments performed under various stress combinations have revealed similar volumetric and deviatoric creep strain behaviour. This situation leads to stress/strain rate relationships of an analogous elastic theory. Here the normal elastic modulus, E, is replaced by $1/\phi(T)$ and the elastic Poisson's ratio, ν, is replaced by its creep counterpart ν_c. Hence we obtain the following Maxwell strain rate equation in three dimensions:

$$\{\dot{\varepsilon}\} = \left(\frac{D}{E} + \phi(T)\right) [V] \{\sigma\} \qquad (1)$$

where at working stress levels $\nu=\nu_c$, D is the differential operator in pseudo-time, D=d( )/dc and

$$[V] = \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix}$$

Virtual Power Principle - For a material which exhibits creep and elastic behaviour, the following relationship exists, at any instant, between the external loads and displacements, and the internal stresses and strains (England 1968):

$$\int_V \{\dot{\varepsilon}\}^T\{\sigma\}dV = \int_S \dot{u}_r P_r dS \tag{2}$$

where $\{\dot{\varepsilon}\}$ and $\{\sigma\}$ refer to the internal strain rates and stresses respectively. P_r are the external forces and $\dot{u}_r$ are the corresponding displacements and

$$\dot{u}_r P_r = \dot{u}_s R_s + \dot{u}_e P_e \tag{3}$$

where P_e and R_s are the applied loads and support reactions, and $\dot{u}_e$ and $\dot{u}_s$ are their corresponding displacement rates. By applying simultaneous small variations to the stresses, strain rates, surface displacement rates at the load points and the reactive forces, it may be established that:

$$\int_V \{\dot{\varepsilon}\}^T\{\delta\sigma\}dV - \int_S \dot{u}_s \delta R_s \, dS = 0 \tag{4}$$

$$\int_V \{\sigma\}^T\{\delta\dot{\varepsilon}\}dV - \int_S P_e \delta\dot{u}_e dS = 0 \tag{5}$$

Equation (4) may be used in conjunction with a Ritz type of stress representation to obtain the time-history of the stresses in structures of complex shape.

Creep Analyses - where a structure is at all times in equilibrium with the specified loading and compatible with respect to both internal strains and external geometrical constraints, with $\dot{u}_s$=0, equation (4) reduces to:

$$\int_V \{\dot{\varepsilon}\}^T\{\delta\sigma\}\, dV = 0 \tag{6}$$

In a Ritz type representation, the state of stress is specified as a series of terms as given in equation (7), for which the spatial stress distributions $\{\sigma_i\}$ are specified. The problem becomes one of identifying the time functions $\{a_i\}$ associated with each stress distribution. These functions represent the preferred weighting of each spatial stress distribution to give the best match to the exact solutions at all times as dictated by the Virtual Power principle viz. equation (6). Thus

$$\{\sigma\} = \{\sigma_o\} + a_1 \{\sigma_1\} + \text{----} + a_n\{\sigma_n\}$$

$$= \{\sigma_o\} + \{a\}^T[\sigma] \tag{7}$$

where $\{\sigma_o\}$ represents any set of stresses which are in equilibrium with the boundary loading, and $[\sigma]$ consitutes self-equilibrating internal stress distributions. The strain rates, $\{\dot{\varepsilon}\}$, of equation (6) may be represented in terms of the stresses of equation (7) through equation (1) and noting that:

$$\delta\sigma = \frac{\partial\{\sigma\}}{\partial a_i}\,\delta a_i = \{\sigma_i\} \tag{8}$$

We have then,

$$\{\dot{\varepsilon}\} = \left(\frac{D}{E} + \phi(T)\right)[V]\{\sigma\}$$

$$= \frac{1}{E}([V]\ [\sigma])\{\dot{a}\} + \phi(T)([V]\ [\sigma])\{a\} \tag{9}$$

$$+ \phi(T)[V]\{\sigma_o\}$$

From equation (7) and equation (8) we obtain:

$$\int_V \{\dot{\varepsilon}\}^T\{\delta\sigma\}\,dV = \int_V [\sigma]^T\{\dot{\varepsilon}\}\,dV = 0 \tag{10}$$

After substituion of $\{\dot{\varepsilon}\}$ from equation (9), equation (10) leads to the following matrix differential equation for the time functions $\{a\}$; thus

$$[P]\{\dot{a}\} + [Q]\{a\} + \{R\} = \{0\} \tag{11}$$

where

$$P_{rs} = \int_V \frac{1}{E}\{\sigma_r\}^T[V]\{\sigma_s\}dV$$

$$Q_{rs} = \int_V \phi(T)\{\sigma_r\}^T[V]\{\sigma_s\}dV \tag{12}$$

$$R_r = \int_V \phi(T)\{\sigma_r\}^T[V]\{\sigma_s\}dV$$

Equation (12) has the solution:

$$\{a\} = e^{-[P]^{-1}[Q]c}\{H\} - [Q]^{-1}\{R\} \tag{13}$$

Where $\{H\}$ is a vector of coefficients determined from the initial condition. This equation may be evaluated readily for any values of pseudo-time, c, to give the weighting functions $\{a\}$ at these times.

When temperatures vary in a cyclic manner in time, a simple modification to the above procedure allows corresponding

stresses to be evaluated. With careful selection of the self-equilibrating stress distributions $\{\sigma_i\}$ the cyclic problem may be solved by simply changing one or more of the $\{a\}$ parameters at the time of each temperature change, by a known amount. This allows a new $\{H\}$ vector of equation (13) to be determined and hence a new transient stress behaviour to be evaluated after each change of temperature, from what is essentially the start of a new initial value problem.

Pseudo-Elastic Analyses - The choice and nature of the self-equilibrating stress distributions are important. VPCREEP has the ability to generate self-equilibrating stress distributions automatically. They are generated from either thermal or non-homogeneous elastic analysis.

FINITE ELEMENT MODEL

The elements used in VPCREEP are 8-nodes for two dimensional (plane stress, plane strain and axisymmetric) problems and 20-nodes for three dimensional problems. They belong to the isoparametric elements of the serendipity family (Zienkiewicz 1982). The assumed displacement field, u_j, being the displacement in the direction j, has the form:

$$u_j = \sum_{i=1}^{n} N_i u_{ji} \qquad j = 1,2,3. \tag{14}$$

where u_{ji} is the value of u_j at the point i and the shape functions N_i are of standard form.

PROGRAMMING REMARKS

The VPCREEP program is not only simple to use but is also very efficient numerically. The following remarks can be made:

- A direct solution algorithm is adopted in contrast to the conventional incremental approach used by most finite element systems.

- The stiffness matrices are formed and reduced only once for elements of the same shape and properties.

- To solve the equation system the frontal solution technique is utilised (Irons 1970).

- Dynamic field length allocation (Overill 1981), i.e. small problems need less field length and will therefore be treated with a higher priority and lower costs in multiple-programming mainframes, such as CDC 7600 at ULCC.

- Restart facility enables the user to enter the program at

any significant point for data checking and additional solutions.

- Versatile input possibilities. Input data may be read from a user defined input file, or alternatively the user may use the program's two and three dimensional mesh generation preprocessor.

- Diagnostic preprocessors are also built in for error detection and early warning before entering the main solution processors.

- Machine independence. VPCREEP is written in standard Fortran and may be installed on a variety of machines. In fact VPCREEP has been run on machines ranging from CDC mainframes to mini-computers.

- Graphic postprocessor may be added without much difficulty.

The operation of VPCREEP system may be considered in four distinct phases and they are illustrated diagrammatically in Figure 1. A simple example of a generated finite element mesh is shown in Figure 2.

BENCHMARK TEST AND NUMERICAL EXAMPLES

Prestressed Wall Element - A prestressed concrete wall element which is subjected to continuous cooling and heating and restrained from thermal warping is analysed as a plane stress problem using 12 8-nodes iso-parametric elements. such conditions are common in many structures, such as oil storage containers, nuclear reactor pressure vessels, pipes and culverts. Figure 3 shows details of the structure and loading conditions. The results are presented in Figure 4. During such continuous cooling and heating, stress reversal takes place and this may introduce extensive cracking if the structure is not designed to sustain cooling and heating of this kind.

Long Thick Cylinder - A long thick cylinder subjected to a sustained through-the-wall temperature crossfall is studied as an axisymmetric problem with 24 8-nodes iso-parametric ring elements. Figure 5 shows details of the cylinder and its loading conditions. The results are illustrated in figure 6. The considerable redistribution of stresses in the structure demonstrates that the combined effect of creep and temperature must be accounted for in design.

In the above two examples, two self-equilibrating stress distributions were used. VPCREEP also has three dimensional

creep analysis capabilities.

CONCLUDING REMARKS AND ACKNOWLEDGEMENTS

The VPCREEP finite element program is designed as a special purpose computer code for the analysis of concrete structures subjected to the combined influence of creep, sustained and time-varying temperatures. The underlying theory is simple to understand and yet practical. VPCREEP can be used either before the design stage, during the design stage or after the design has been completed because it is inexpensive to use. To use the program is also a very simple matter even for relatively inexperienced sporadic users because most of the processes are automatic.

Looking into the future, it is planned to extend the element library and to produce a vectorised version for implementing on the vector oriented machines.

The authors wish to thank the financial support given by the Health and Safety Executive. We are also indebted to Dr. R.E. Overill of the Computer Centre, King's College London for his invaluable advice and help. We would also like to express our appreciation to Miss Anne Alexandre who typed this paper for us.

REFERENCES

Barsoum, R.S., Loomis, R.W. and Steward, P.D. (1976) ASME-MPS Symposium on creep-fatigue interaction. ASME, New York p.67.

Dhalla, A.K. and Roche, R.V. (1975) Advance in design for elevated temperature environment. ASME, New York p.83.

England, G.L. (1966) Steady state stresses in concrete structures subjected to sustained loads and temperatures, Part I and II. Nucl. Eng. Des. 3, 54, p. 246.

England, G.L. (1967) Numerical creep analyses applied to concrete structures. ACI Journal.

England, G.L. (1968) Time-dependent stresses in creep-elastic materials - A general method of calculation. JBCSA, Conference on recent advances in stress analysis: New concepts and their practical application.

England, G.L. (1968) Long-term thermal stresses in prestressed concrete structures. Proc. ICE conf. prestressed vessels, London.

England, G.L., Labib, G.A. and Moharram, A. (1977) The influence of time and temperature on the working stresses in concrete structures. Research Report No. 48, Cement and Concrete Assoc.

Irons, B.M. (1970) A frontal solution program for finite element analysis. Int. J. Num. Meth. Engng. 2., No. 1.

Krishnamoorthy S., England, G.L. and Yu, C.W. (1971) The behaviour of prestressed concrete portal frames as influenced by creep and temperature. MCR, Vol. 23, No. 74.

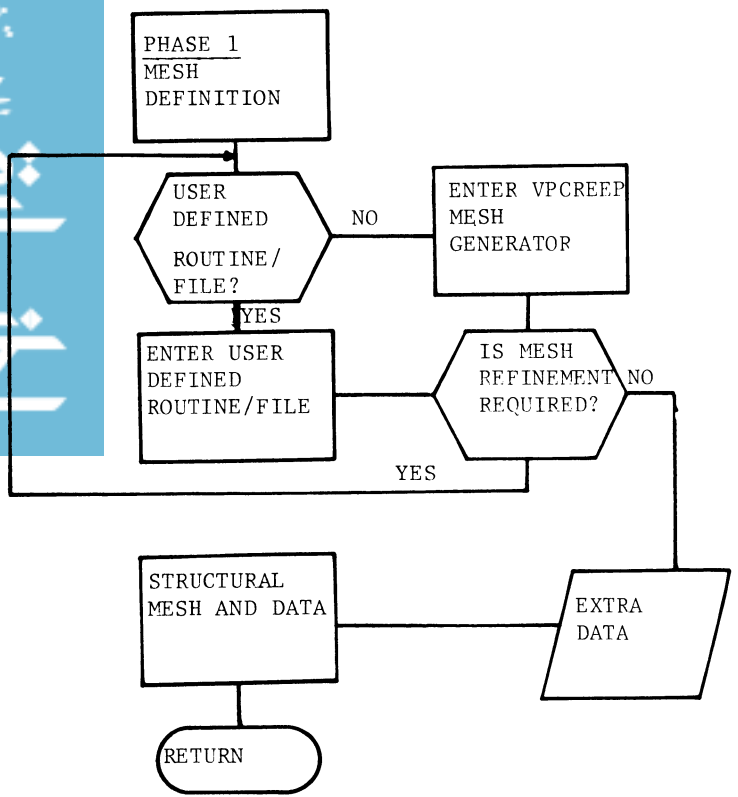
Overill, R.E. (1981) Extension of SCM and LCM field length on the CDC 7600. King's College London Computing bulletin K3.2/8 and private communication.

Ross, A.D., England, G.L. and Suan, R.H. (1965) Prestressed concrete beams under a sustained temperature crossfall. MCR Vol. 17, No. 52.

Ross, R.D. (1958) Creep of concrete under variable stress. J. Am. Conc. Inst. Proc. 54 (9).

Suan, R.H. (1964) The long-term behaviour of simply-supported and continuous concrete beams subjected to a thermal gradient. Ph.D. thesis University of London.

Zienkiewicz, O.C. (1982) The finite element method. 3rd ed., McGraw Hill.



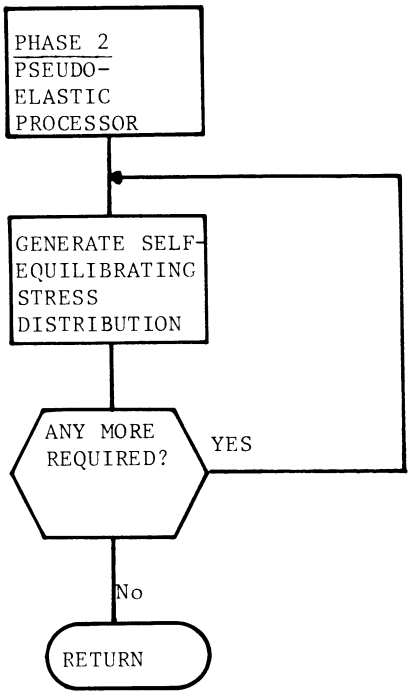
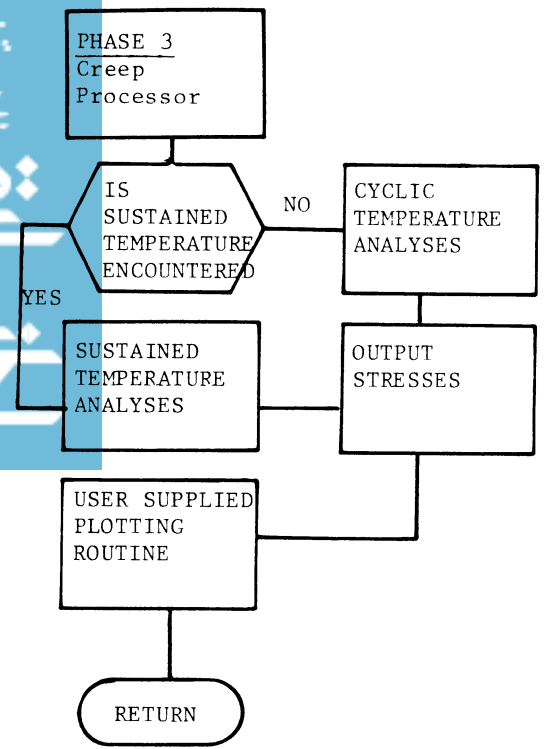


Figure 1a: VPCREEP Overview - Phase 1 and 2.



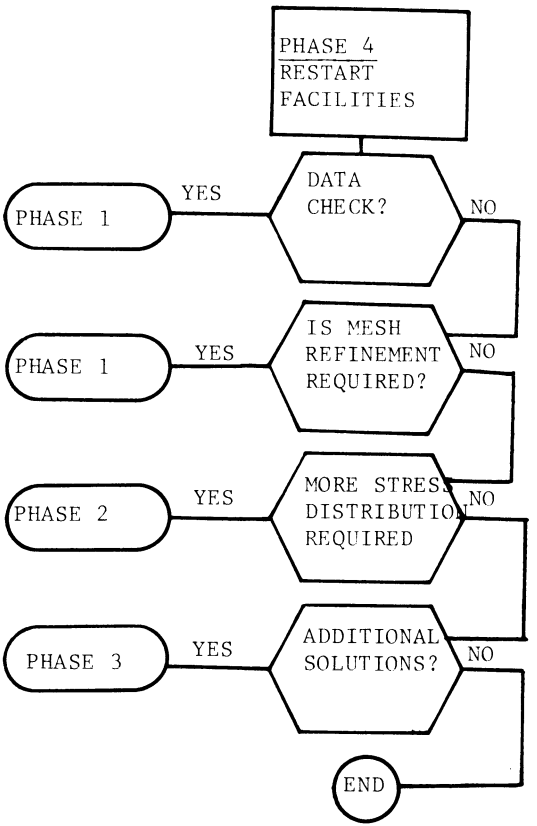


Figure 1b: VPCREEP Overview - Phase 3 and 4.

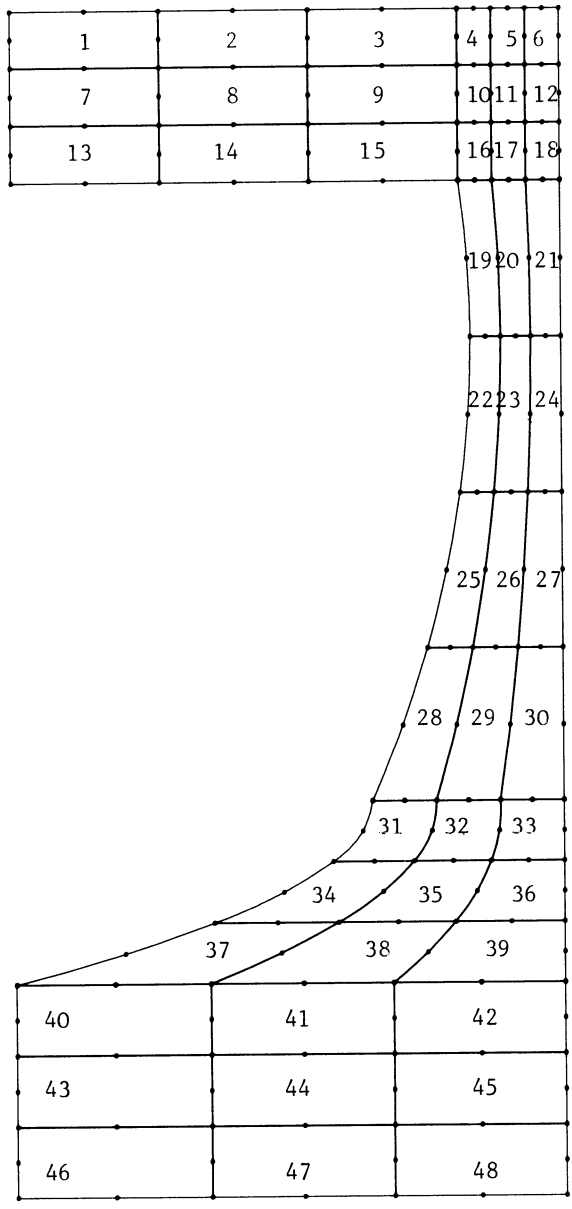


Fig. 2: A Simple Finite Element Mesh Generated by VPCREI (Nodal Numbers Not Shown)

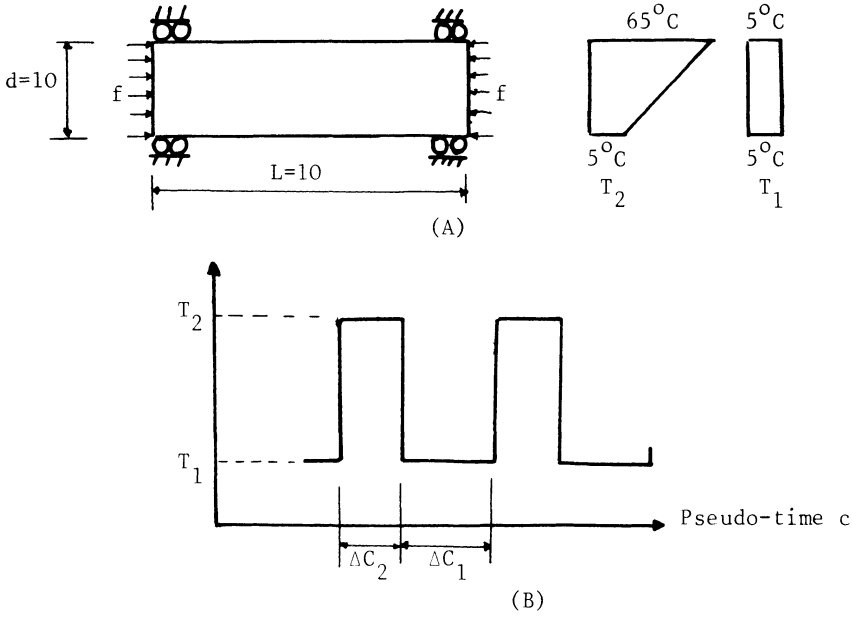


Figure 3: (A) Structural Configuration and Loading Conditions.
(B) Nature of Temperature Cycling.
(C) Finite Element Idealization. (Nodal Numbers not shown)

Data Used: $f = 7\ MN/m^2$
$E = 34\ GN/m^2$
$\alpha = 12 \times 10^{-6}/^oC$
$\nu = 0.2$
c in MN/m^2 per ^oC

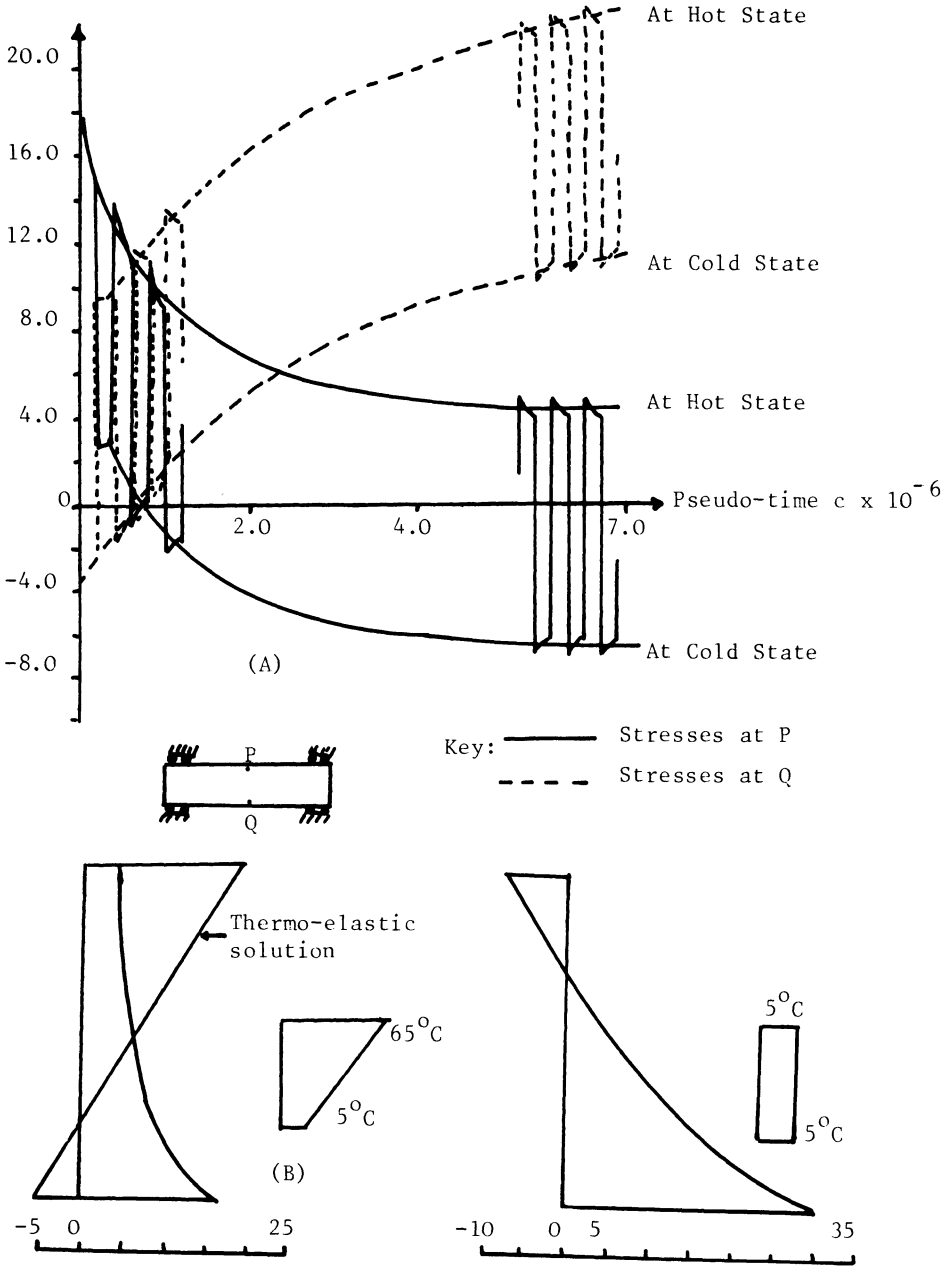
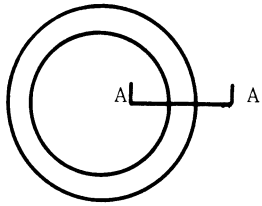


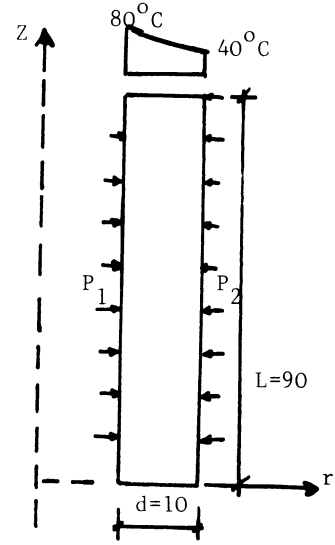
Figure 4: (A) Time-dependent Variation of Stresses in Outer Fibres
(B) Steady-State-Cyclic Stresses.

Stress in MN/m^2 +ve Compression; -ve Tension
Pseudo-time in MN/m^2 per ^{o}C.

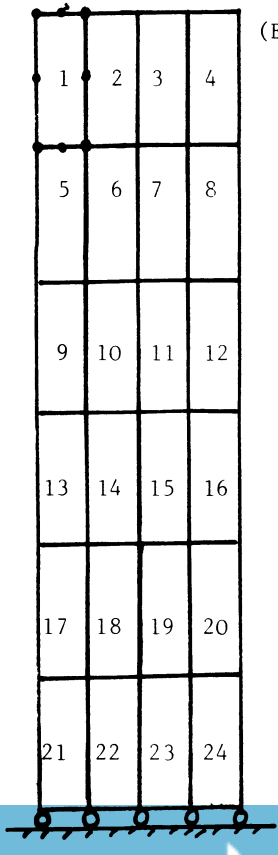


Plan

(A)



Vertical Section
through A-A



(B)

Data Used:

$\frac{r_o}{r_i} = 2$

r_o External radius

r_i Internal radius

P_2 External pressure

P_1 Internal pressure

$T_r = 80^oC - 57.7 \text{ Ln } (\frac{r}{r_i})\ ^oC$

$P_1 = 3.45\ MN/m^2$

$P_2 = 1.38\ MN/m^2$

$E = 34\ GN/m^2$

$\nu = 0.2$

α = Coefficient of thermal expansion
$= 12 \times 10^{-6}/^oC$

c in MN/m^2

Cylinder restrained in Z-direction

Figure 5: (A) Structural Configuration and Loading Conditions.
(B) Finite Element Mesh (Nodal numbers omitted)

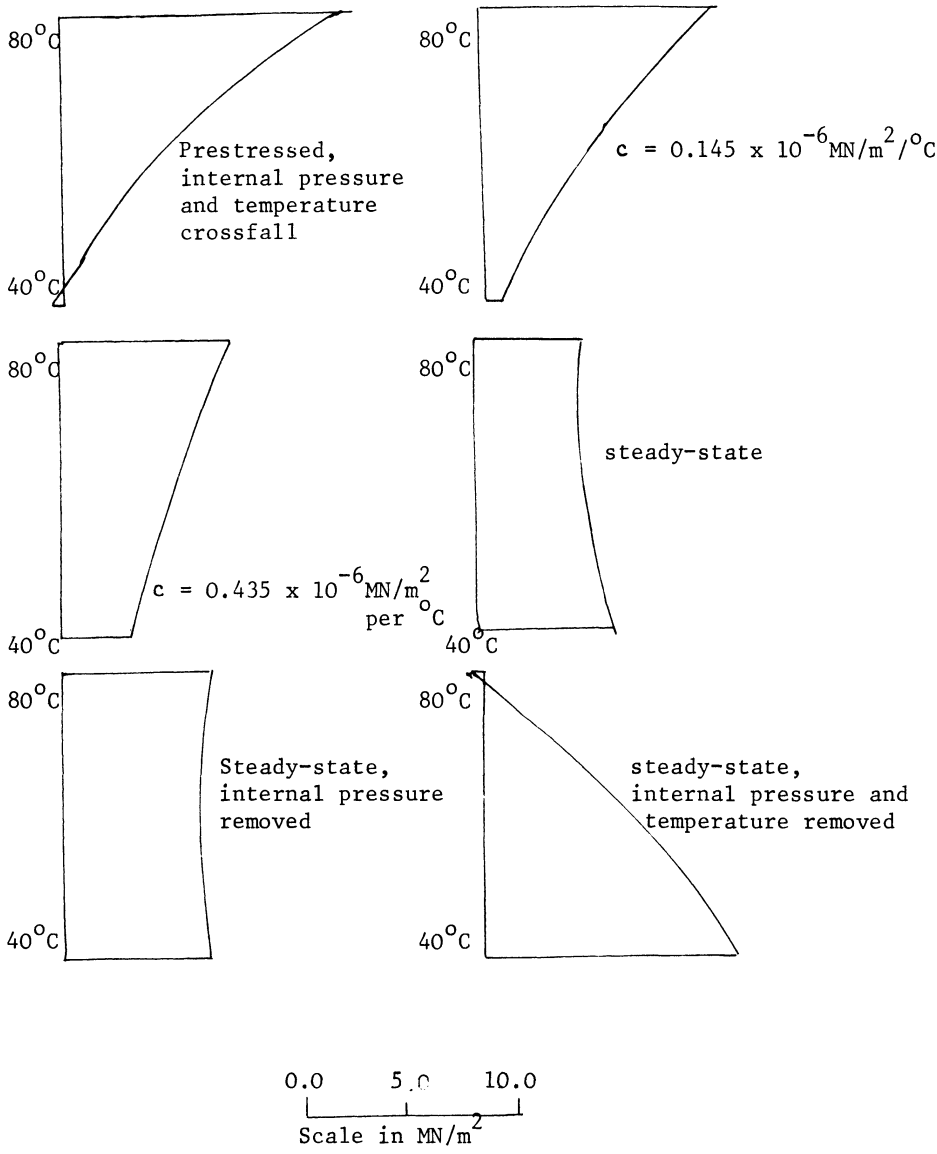


Figure 6: Circumferential Stresses for Long Thick Cylinder.

STRUCTURAL ENGINEERING

AN EXTENSION OF THE CONTINUOUS MEDIUM METHOD FOR INTERCONNECTED SHEAR WALLS

A.Cholewicki
Centre for Building Systems Research and Development, Warsaw

M.Winiarski
Technical University of Warsaw

INTRODUCTION

The analysis of interconnected shear walls with regular rows of openings, through an easy task at first glance, presents specific difficulties regarding computation methods. The reason for this is that interconnected walls form a combination of members, some of which should be treated as plates and some as beams. Of the three groups of methods available for the solution of such problems:

- Finite Element Method
- Modified Frame Analysis /Mac Leod, 1973/
- Continuous Medium Method /Rosman, 1964/,

the first one is obviously uneconomic for this purpose
In order to achieve the required degree of accuracy, the lintels have to be modelled using several elements, and-in consequence- the number of elements in a wall panel one storey high with one opening varies from 10-15 in the case of higher order elements to 30-40 simple, constant strain elements. Ordinary frame programs, available on most computers, require several modifications to produce reliable results in the case of interconnected shear walls. Most members are far from slender; node dimensions are large and cannot be neglected as in the ordinary frame theory. They can be modelled using the wide column frame method, i.e. by introducing infinitely stiff lintel ends between the edges of the openings and the wall axes This does not, however, allow for the elastic encastering of the lintel in the wall. To correct this, the stiff lintel ends can be made somewhat shorter; there are several formulae to arrive at this result /Michael, 1967/, /Cholewicki, 1973/. For some ratios of lin-

tel depth to wall width the solution is very sensitive to the smallest changes in lintel dimensions and errors up to 15% are possible /Winiarski, 1974/ In a majority of frame programs, the effect of shear deformations is neglected and this can result in errors exceeding 10% when the ratio of member depth to length is over 0.3. As most of the frame programs allow prismatic members only, the application of wide column frame method doubles the number of nodes in a frame with the consequent increase in the volume of data and the computer costs.
There exist systems and specialised programs in which all these facilities are available, but this is not the case with the programs encountered most frequently.
The main assumption of the last of the discussed methods is to replace the discrete interconnection /lintels/ by a continuous medium of equivalent elastic properties /see Fig 2/. This does not produce substantial errors as long as the number of lintels exceeds 5 or 6 and the ratio of lintel depth to wall width is small.
The unknowns are the unit shears acting in the connections. Their number is equal to the number of the rows of openings and is very small when compared with the F.E. or Frame Methods The small size of matrices /30 by 30 to 50 by 50 for an average cross-wall building/ allows an in-core solution, resulting in extremely short computing times. Although the Ritz energy method used in the solution of the continuous medium model leads to symmetrical but full matrices, this is of minor importance in view of their small dimensions. Small core requirements and short computing times make the Continuous Medium Method programs more design-oriented than programs based on the other two methods discussed above.
"The distinction between analysis and design is important There is a common tendency among many engineers to assume that a structural analysis capability is synonymous with the corresponding design ability. This is, in fact, not so." /B.Faulkner, 1973/.
The main disadvantage of the Continuous Medium Method are the rigid restrictions imposed upon the shape of the wall and types of loading. The dimensions and material constants cannot change throughout the entire height. In most papers on the analysis of plane walls and in all papers known to the authors, on the three-dimensional analysis of wall systems the above restrictions are observed In the design practice, the difficulties arise when chan-

ges occur in ground floor plans of multi-storey buildings due to functional requirements, resulting frequently in larger openings or in replacing of some walls by columns. Similarly, in tall buildings, this method cannot allow for the variations in wall thickness or in the quality of concrete
The development, presented below, and the computer program based on this algorithm do not only eliminate these disadvantages of the Continuous Medium Method, making it far more flexible, but also improve its accuracy.

CONTINUOUS MEDIUM METHOD AND ITS EXTENSION

Assumptions of the C.M. method

- Plane cross sections remain plane after the deformations have taken place
- No local rotations of wall axes occur at the intersections with lintel axes
- Shear deformations are neglected
- Walls are rigidly fixed in the foundation
- Lintels have infinite longitudinal stiffness /i. e. wall axes remain equidistant/
- Lintels are treated as beams; shear deformation and elastic encastering in the walls are taken into account /see Fig. 1/

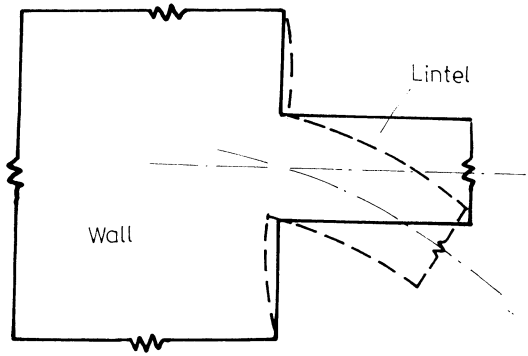


Figure 1. Elastic encastering of lintel in wall

Assumptions made in the extension of the C.M.Method

- Walls are divided horizontally into zones having different geometrical and material properties, constant within a zone
- The properties of lintels /dimension and material/ can very between the zones but remain uniform within a zone

The solution of the problem consists in defining

the function T (x) which gives the magnitude of the shear force in the connecting medium throughout the height of the building.
For walls with more than two rows of openings the general solution is difficult to obtain, and approximate methods are used instead. The greater flexibility of the presented method as well as its greater accuracy can be understood at a glance in Fig 2.
Fig. 2a shows the basic approximations of the conventional method of solution, used in most programs. The wall is uniform; the unit shear distribution is approximated by its average /rectangle/ which is far from reality. The resulting total shear, which is the summation of the unit shears is approximated by a triangle, which is fairly close to the curve representing "the exact" result.

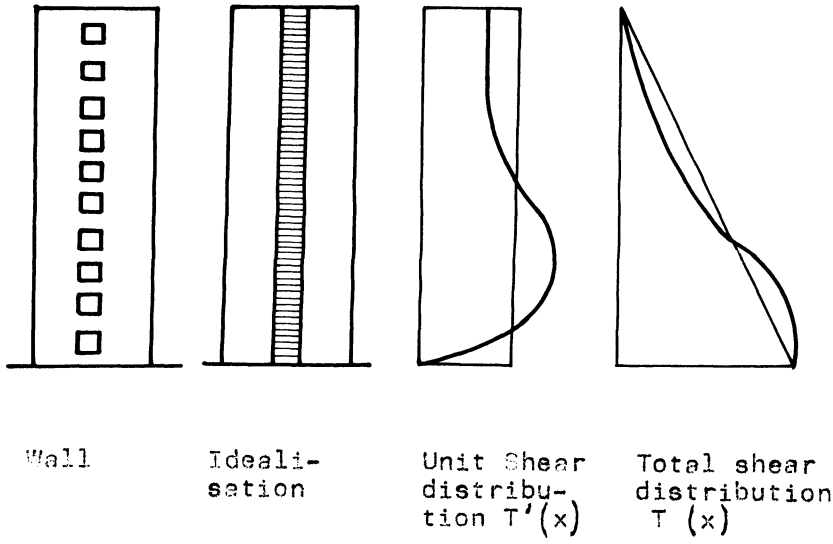


Figure 2a The Continuous Medium Method

Fig. 2b shows the basic advantages of the new development. The wall need not be uniform throughout its height; the approximation of unit shear distribution is far more exact, which influences favourably the exactness of the resulting total shear and remaining internal forces.

The algorithm of the method

As mentioned above, in the newly developed method, the distribution of shear throughout the height of a row of connections is approximated by average shears in each of the zones

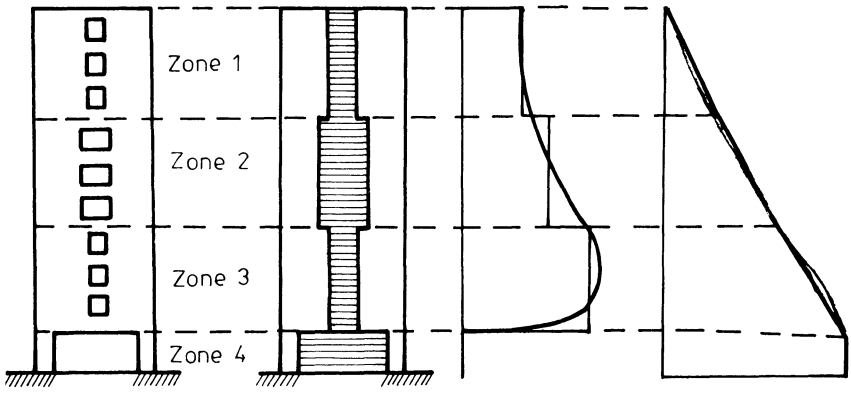


Figure 2b Extension to Continuous Medium Method

This approximation leads to the following relations between the shear forces in each of the zones in a row of connections:

$$\begin{aligned} T_{1j} &= \sum_{k=1}^{j-1} T^*_{1k} + \frac{X_j}{H_j} T^*_{1j} \\ \cdots \quad & \quad \cdots \quad \cdots \\ T_{ij} &= \sum_{k=1}^{j-1} T^*_{ik} + \frac{X_j}{H_j} T^*_{ij} \\ \cdots \quad & \quad \cdots \quad \cdots \\ T_{nj} &= \sum_{k=1}^{j-1} T^*_{nk} + \frac{X_j}{H_j} T^*_{nj} \end{aligned} \qquad /1/$$

where: H_j - the height of the j^{th} zone
T^*_{ik} - the increment of the shear force in the j^{th} zone of the i^{th} row of connections
X_j - the distance measured from the top of the j^{th} zone
$i = 1,2 \ldots n$ - row counter
$j,k = 1,2 \ldots s$ - zone counters

Generally, the first subscript refers to the row number and the second - to the zone number
The solution of the problem consists in finding the values of T_{ij} via the principle of minimum energy

The energy stored in the structure:

$$2U = \int_0^{H_1}\Bigg[\frac{\left(T'_{11}\right)^2}{C_{11}} + \ldots \frac{\left(T'_{n1}\right)^2}{C_{n1}} + \frac{T_{11}^2}{E_1F_{11}} + \frac{\left(T_{21}-T_{11}\right)^2}{E_1F_{21}} + \ldots \frac{\left(-T_{n1}\right)^2}{E_1F_{(n+1)1}} +$$

$$\ldots + \frac{1}{E_1I_1}\left(M_1^o - T_{11}L_{11} - T_{21}L_{21} - \ldots - T_{n1}L_{n1}\right)^2\Bigg]dX_1 +$$

$$+ \int_0^{H_j}\Bigg[\frac{\left(T'_{1j}\right)^2}{C_{1j}} + \ldots \frac{\left(T'_{nj}\right)^2}{C_{nj}} + \frac{T_{1j}^2}{E_jF_{1j}} + \frac{\left(T_{2j}-T_{1j}\right)^2}{E_jF_{2j}} + \ldots \frac{\left(-T_{nj}\right)^2}{E_jF_{(n+1)j}} +$$

$$+ \frac{1}{E_jI_j}\left(M_j^o - T_{1j}L_{1j} - T_{2j}L_{2j} - \ldots - T_{nj}L_{nj}\right)^2\Bigg]dX_j \ldots$$

$$\ldots + \int_0^{H_s}\Bigg[\frac{\left(T'_{1s}\right)^2}{C_{1s}} + \ldots \frac{\left(T'_{ns}\right)^2}{C_{ns}} + \frac{T_{1s}^2}{E_sF_{1s}} + \frac{\left(T_{2s}-T_{1s}\right)^2}{E_sF_{2s}} + \ldots \frac{\left(-T_{ns}\right)^2}{E_sF_{(n+1)s}} +$$

$$+ \frac{1}{E_sI_s}\left(M_s^o - T_{1s}L_{1s} - T_{2s}L_{2s} - \ldots - T_{ns}L_{ns}\right)^2\Bigg]dX_s$$

/2/

where: C_{ij} - stiffness of connection /lintel or joint/ in the i^{th} row of the j^{th} zone

E_j - Young's Modulus of the wall in the j^{th} zone

F_{ij} - cross sectional area of the i^{th} wall in the j^{th} zone

M_j^o - total bending moment due to the external load, at the lower end of the j^{th} zone

/$M_j^o = 1/2\ w \cdot \left(\sum H_j\right)^2$, where w - uniformly distributed horizontal load per unit of the height/

I_{ij} - moment of inertia of the i^{th} wall, in the j^{th} zone

T_{ij} - total shear in the i^{th} row at the lower end of the j^{th} zone

T'_{ij} - the derivative of T_{ij}

L_{ij} - the distance between the centres of gravity of the i^{th} and the $i-1^{th}$ wall, j^{th} zone

The remaining symbols have been already defined.
On substituting from /1/ into /2/ and integrating,

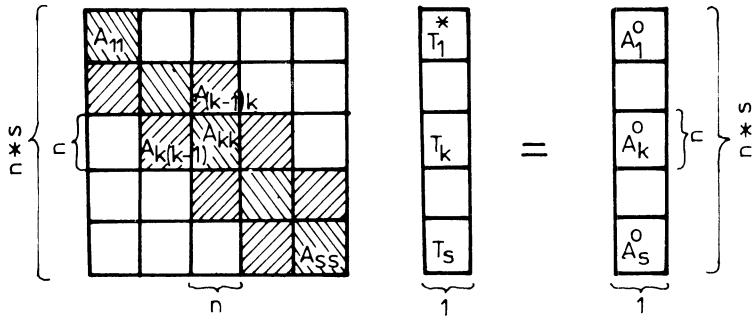
the energy of the system is obtained in the form of a second degree function of T^*_{ij}. The unknown parameters T^*_{ij} can be found using the principle of the minimum energy:

$$\frac{\partial(2U)}{\partial T^*_{ij}} = 0 \qquad /3/$$

The differentation of the expression for the energy results in a set of linear algebraic equations:

$$[A]\{T^*\} = \{A^o\} \qquad /4/$$

Square matrix $[A]$ is composed of sub-matrices $[A_{kL}]$ vectors $\{T\}$ and $\{A^o\}$ are composed of sub-vectors $\{T_L\}$ and $\{A^o_L\}$ as follows:



The dimensions of sub-matrices $[A_{KL}]$, are $n * n$, and those of the sub-vectors - $n * 1$
Each of the sub-matrices $[A_{KL}]$ is a symmetrical, full matrix with the elements on the diagonal computed according to one formula, those on the sub-diagonal-according to the other and the remaining elements- using the third formula, The same applies to the sub-matrices on the diagonal, sub-diagonal and the remainder, so that the elements of the matrix $[A_{ij}]$ are computed using nine different formulae. Matrix $[A]$ can be considered as an equivalent of the Flexibility Matrix in the classical methods; vector $\{T^*\}$ is the vector of the unknowns and vector $\{A^o\}$ can be considered as an equivalent of the load vector in the classical methods.
The internal forces and moments are computed starting with the shear force at the boundary between two zones:

$$T_{ij} = \sum_{k=1}^{j} T^{*}_{ik} \qquad /5/$$

Total bending moment at the same boundary:

$$M_j = \frac{1}{2} w \left(\sum_{k=1}^{j} H_k\right)^2 - \sum_{i=1}^{n} T_{ij} L_{ij} \qquad /6/$$

where M_j is the total bending moment in connected walls, in the j^{th} zone.
Since we have assumed equal deflections of all walls, this moment is distributed in proportion to wall stiffnesses

$$M_{ij} = \frac{I_{ij}}{I^{o}_{j}} M_j \qquad /7/$$

The longitudinal forces in the walls, at the same boundary:

$$N_{ij} = T_{ij} - T_{(i-1)j} \qquad /8/$$

and the shear force in the lintels:

$$Q_{ij} = \frac{T^{*}_{ij}}{H_j} \cdot h \qquad /9/$$

where h is the storey's height
The horizontal deflections, δ_r, of an interconnected wall system can be computed at the top of the wall and at all boundaries between the zones via the principle of the virtual work.

$$\delta_r = \sum_{j=1}^{s} \frac{1}{E_j I^{o}_{j}} \left\{ \frac{wH_j}{2} \left(\sum_{k=1}^{j-1} H_k\right)^2 \left(\sum_{k=r}^{j-1} H_k\right) + \frac{wH_j^2}{2} \left(\sum_{k=1}^{j-1} H_k\right) \left(\sum_{k=r}^{j-1} H_k\right) + \right.$$

$$+ \frac{wH_j^3}{6} \sum_{k=r}^{j-1} H_k - H_j \sum_{k=r}^{j-1} H_k \left[\sum_{i=1}^{n} \left(T_{i(j-1)} + \frac{T^{*}_{ij}}{2} \right) L_{ij} \right] + \frac{wH_j^2}{4}$$

$$\left. \left(\sum_{k=1}^{j-1} H_k^2\right) + \frac{wH_j}{3} \sum_{k=1}^{j-1} H_k + \frac{wH_j^4}{8} - H_j^3 \sum_{i=1}^{n} \left(\frac{T_{i(j-1)}}{2} + \frac{T^{*}_{ij}}{3} \right) L_{ij} \right\}$$

/10/

COMPUTER PROGRAM

User aspect

The present version of the program allows the user to carry out statical analyses of buildings having up to 60 rows of interconnections and 20 zones throughout the height having different characteristics with the proviso that the product of these two numbers cannot exceed 150.

The walls may include stiffening walls /or their active parts/ at right angles to the main wall, as shown in Fig. 3.

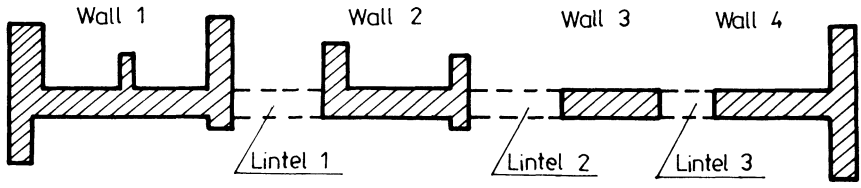


Figure 3. Horizontal section through interconnected walls.

The method is appicable not only to walls interconnected by regular rows of lintels, but also to walls built-up of precast panels /with or without rows of lintels/, treating the joints as rows of interconnections. Fictitious joints can be introduced in monolithic walls, assigning them appropriate stiffnesses /Cholewicki, 1980/ An example can illustrate these points.

A plane wall with irregular openings is shown in Fig. 4. Fictitious joints are positioned on either side of the openings. The strips of the wall are given appropriate moments of inertia. Where the openings occur, these are equal to zero.

The stiffness of the fictitious joints is equal to the stiffness of the lintel in the zones with openings, and to the stiffness of the wall where there are no openings. In zone 7 the stiffnesses of the joints are zero, and so are the moments of inertia of wall strips 2,3 and 4. Zone 6 was introduced to allow stress concentrations above the large opening to be exposed.

In this way, the method approximates the Finite Strip Method.

/No of joints/ * / No of zones/ = 4 * 7 = 28.

Software aspect

The computer program is written mainly in ANSI FORTRAN, for an ICL 1900 series. It contains a small, 20-line master and five overlay modules, not exceeding 150 lines each.
First module reads and checks the data.
The Second carries out all preliminary calculations needed for the elements of sub-matrices
The third module sets up the sub-matrices and the global matrix. The solution is carried out by the fourth module, using Gauss algorithm for symmetrical matrices
Last overlay takes from the module No. four the unit shears, which are the result of the solution of the system of equations, carries out further computations and prints unit shear, longitudinal forces, bending moments, stresses and deflections, as mentioned in user aspect.

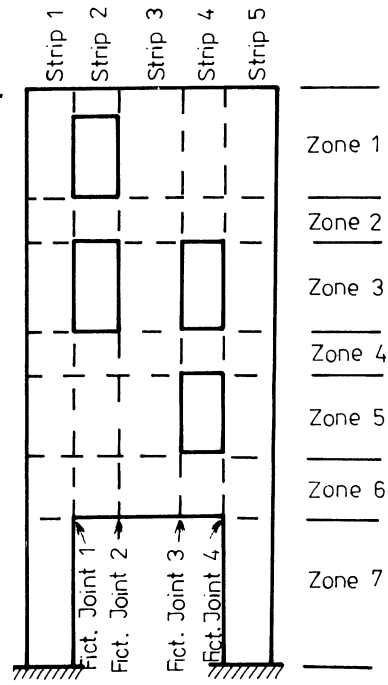


Figure 4 Wall with ir regular openings

The program requires less than 12 K words of space plus 28 K working space in the core The solution of the system of equations is in-core for problems up to 150 unknowns and in several blocks when the number of unknowns is greater.
The computing time is extremely short: several seconds C.P.U. and a couple of minutes START to STOP.
The data consist of the geometrical description of the plan at every zone; next, lintel stiffnesses and Young's Moduli for every zone are input, as well as the vector of zone heights and the storey height. Load data include a single value for uniformly distributed horizontal load and a vector of vertical point loads /maximum- one load per wall/. The vertical loads, applied at the top of the building are assumed to act at the centres of gravity of wall sections.
The result include: geometrical properties of wall sections /area, centre of gravity, moment of iner-

tia/, unit shear forces, total shear forces, longitudinal forces and bending moments in walls as well as longitudinal stresses at the edges of the walls; shear forces in lintels and a vector of horizontal deflections at the boundary of each zone.

NUMERICAL EXAMPLE

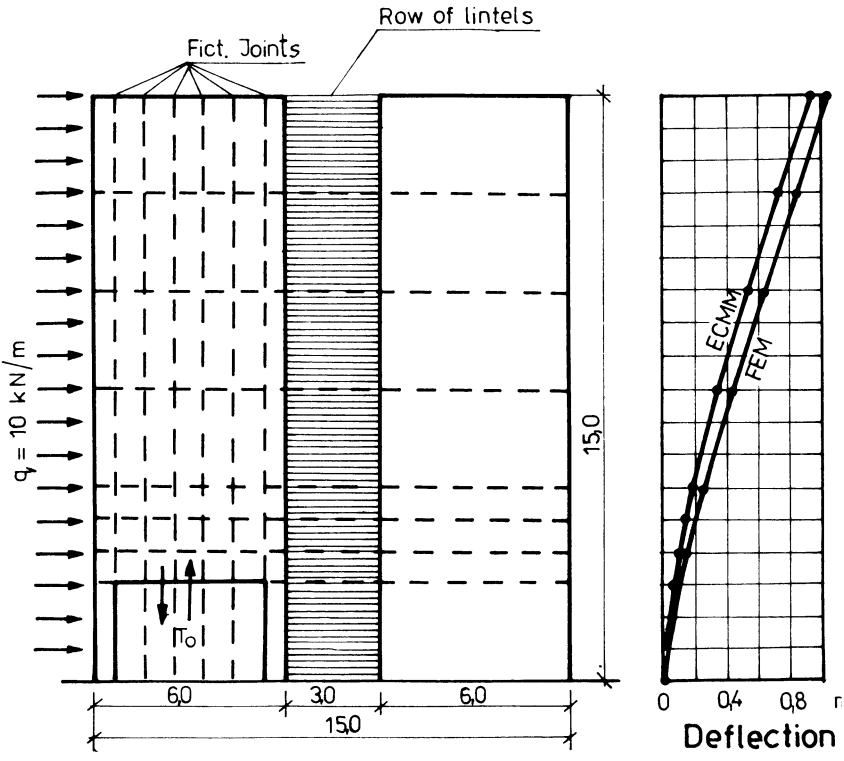


Figure 5a. Interconnected shear wall and the computed deflection.

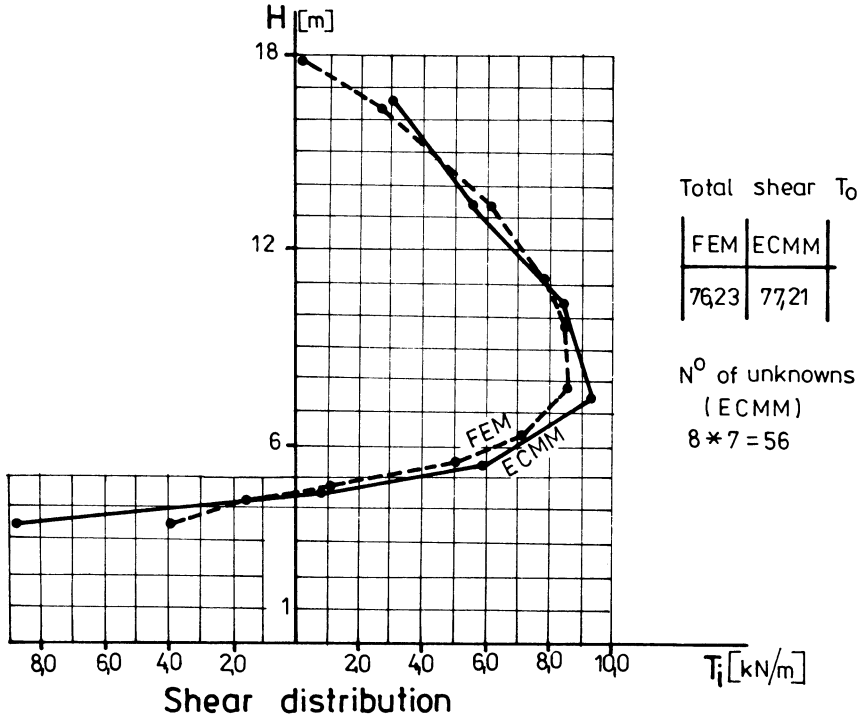


Figure 5b. Comparison of the results by FEM and ECMM

References

Cholewicki,A.,Makulski,W.,Winiarski,M.(1973)Analysis of shear walls with rows of openings /in Polish/.InformatorProjektanta Bud.Ogólnego 1: 3 - 40

Cholewicki,A.,(1980)Analysis and design of shear walls /in Polish/.Centre for Building Systems R.D.227pp

Faulkner,B.,Adamchak,J.G.,Snyder,G.J.,Vetter,M.F. (1973)Synthesis of welded grillages to withstand compression and normal loads.Computers and Structures 2: 221 - 246

MacLeod,I.,(1973)Analysis of shear wall buildings by the frame method.Proc.I.C.E.Part 2,9 : 593 - 603

Michael,D.,(1967) The effect of local wall deformation on the elastic interaction of cross walls coupled by beams.Tall Buildings.Pergamon Press,253 - 270

Rosman,R.,(1964) Approximate analysis of shear walls subjected to lateral loads.Journal A.C.I. 61: 717 - 732

Winiarski,M.,Krol,A.(1974) Statical analysis of coupled shear walls by the modified frame method /in Polish/ XXth Conference of the Polish Acad. of Sci. Civ. Eng.Committee,1: 301 - 309

BAVOMU : A COMPUTER PROGRAM DEVOTED TO MULTIPLE-ARCH DAMS

R.E. Arnould, University of Liege (Belgium)

INTRODUCTION

Generally, multiple-arch dams are constituted by a large number of cylindrical raked shells supported by parralel buttresses, showing a triangular or trapezoidal shape, and two-dimension strain conditions (Figure 1).

The designer must choose the main following items :

H : height of the dam	L : distance between buttresses
β : raking angle of the vault	ρ : curvature radius of the vault
δ : thickness of the vault	δ_c : thickness of the buttress

The calculation of the stresses and strains of such a structure was solved first considering the shells and the plates first separately, taking account of their own loadings and specific linkages, then by considering the whole of the structure (Fig. 2)

A good pattern of the strain and stress fields inside the vaults as along their boundaries is obtained by using the classical Flügge's equations written for cylindrical symmetric shells (Flügge, 1960).

CHANGE OF VARIABLES

However, the above-mentioned equations are suitable for orthogonal axis (Figure 3).
Yet most dams have bounds along two horizontal parallel planes (Figure 4). In those conditions, the numerical integration of differential equations through an orthogonal grid gives many problems along the boundaries. Therefore, we use skew axis with a curvilinear grid (Figure 5).

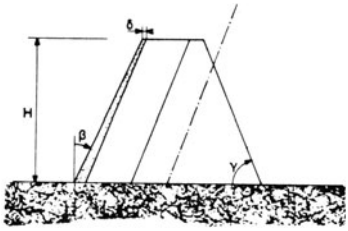
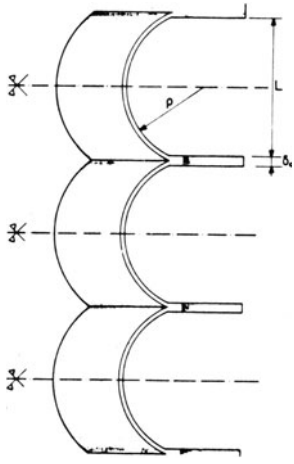


Figure 1. - *Cross Sections of a multiple arch dam*



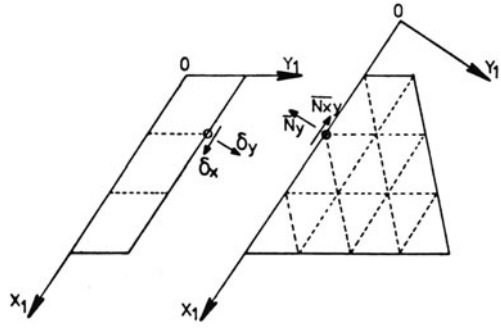


Figure 2. - *Linking elements between vault and buttress*

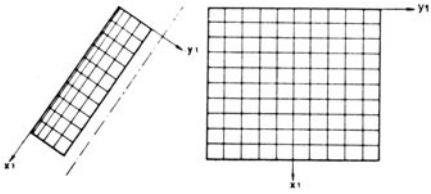


Figure 3. - *Raked vault with orthogonal bounds*

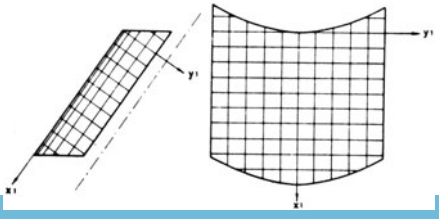


Figure 4. - *Raked vault with horizontal bounds*

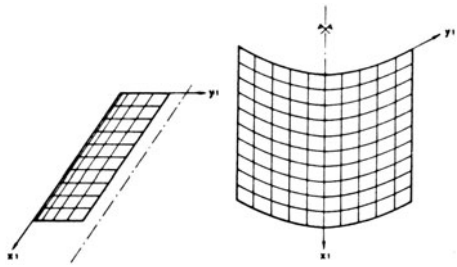


Figure 5. - Raked vault with horizontal bounds (curvilinear net)

Hereunder we write the 3rd equilibrium differential equation related to the Ox1 and Oy1 axis thus defined.

$$\frac{\partial u}{\partial x_1}\left(\frac{1}{\rho}(D\nu-K'')\right) + \frac{\partial^2 u}{\partial x_1^2}\left(-2\,\frac{K'}{\rho}\right) + \frac{\partial^3 u}{\partial x_1^3}\left(-\frac{K}{\rho}\right) + \frac{\partial v}{\partial x_1}\left(\frac{ab}{\rho}\,(-K''\nu+D)\right)$$

$$+ \frac{\partial v}{\partial y_1}\left(\frac{a}{\rho}\,(-K''\nu+D)\right) + \frac{\partial^2 v}{\partial x_1^2}\left(-\frac{ab}{\rho}\,2K'\nu\right) + \frac{\partial^2 v}{\partial x_1 \partial y_1}\left(-\frac{a}{\rho}\,2K'\nu\right)$$

$$+ \frac{\partial^3 v}{\partial x_1^3}\left(-\frac{ab}{\rho}\,K\nu\right) + \frac{\partial^3 v}{\partial x_1^2 \partial y_1}\left(-\frac{a}{\rho}\,K\nu\right) + w\left(\frac{1}{\rho^2}\,(D+K'')\right)$$

$$+ \frac{\partial w}{\partial x_1}\left(\frac{ab}{\rho^2}\,2K'\right) + \frac{\partial w}{\partial y_1}\left(\frac{a}{\rho^2}\,2K'\right) + \frac{\partial^2 w}{\partial x_1^2}\,(K''\nu+K''+2ab(1-\nu)K''$$

$$+ a^2b^2\left(\frac{2K}{\rho^2} + K''+K''\nu\right)) + \frac{\partial^2 w}{\partial x_1 \partial x_1}\,(2a(1-\nu)K'' + 2a^2b\left(\frac{2K}{\rho^2} + K''+ K''\nu\right))$$

$$+ \frac{\partial^2 w}{\partial y_1^2}\,a^2\left(\frac{2K}{\rho^2} + K''+ K''\nu\right) + \frac{\partial^3 w}{\partial x_1^3}\,2(1+a^2b^2)(K'+abK')$$

$$+ \frac{\partial^3 w}{\partial x_1^2 \partial y_1}\,(2aK'(1+3a^2b^2) + 4a^2bK') + \frac{\partial^3 w}{\partial x_1 \partial y_1^2}\,(2a^2(K'+3abK'))$$

$$+ \frac{\partial^3 w}{\partial y_1^3}\,(2a^3K') + \frac{\partial^4 w}{\partial x_1^4}\,K(1+2a^2b^2+a^4b^4) + \frac{\partial^4 w}{\partial x_1^3 \partial y_1}\,4a^2bK(1+a^2b^2)$$

$$+ \frac{\partial^4 w}{\partial x_1^2 y_1^2}\,2Ka^2(1+3a^2b^2) + \frac{\partial^4 w}{\partial x_1 \partial y_1^3}\,(4a^4bK) + \frac{\partial^4 w}{\partial y_1^4}\,(a^4K) = Z$$

(1)

Let us just remark that the first two equilibrium conditions present a slightly more simple form...

BOUNDARY CONDITIONS

Support on the buttress

The significant displacements are sketched on the Figure 6. We express the equality of displacements in both structures and conditions resulting from symmetry

a) $v \sin\phi - w \cos\phi = \delta_y$ (2)

b) $u = \delta_x$ (3)

c) $v \cos\phi + w \sin\phi = 0$ (4)

d) $\delta_\phi = w^\bullet = 0$ (5)

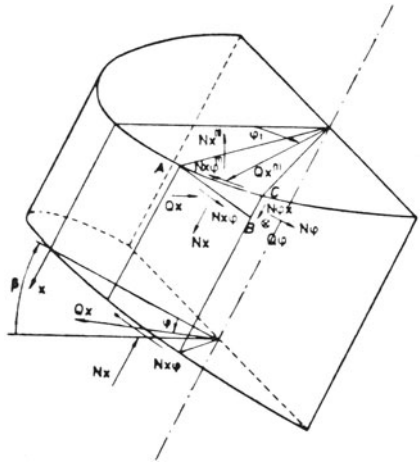


Figure 6. – Displacements along the buttress boundary

Conditions on the crest

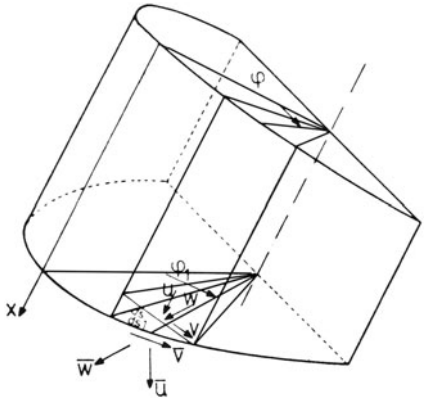
These conditions are derived by expressing the equilibrium of a curvilinear triangular element ABC (Figure 7).

Considering the Force $N_x^{(1)}$, we write :

$$N_x^{(1)} ds_1 = N_x\, ds \cos\beta + N_{x\phi}\, ds \sin\phi \sin\beta + Q_x\, ds \cos\phi \sin\beta + N_{\phi x}\, ds_2 \cos\beta + N_\phi\, ds_2 \sin\phi \sin\beta + Q_\phi\, ds_2 \cos\phi \sin\beta \quad (6)$$

We have demonstrated that :

$$\frac{ds}{ds_1} = \frac{\cos\beta}{\sqrt{\sin^2\phi_1 + \cos^2\phi_1 \cos^2\beta}} = \frac{\cos\beta}{\sqrt{\ell \cos\beta}} \; ; \quad \frac{ds_2}{ds_1} = \sin\beta \sin\phi_1 \quad (7)$$



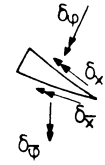


Figure 7. - *Upper boundary Equilibrium*

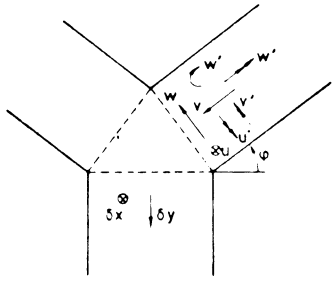


Figure 8. - *Displacements along the foundation*

Hence, we derive a first condition :

$$N_x^{(1)} = N_x \frac{\cos^2\beta}{\sqrt{\ell \cos\beta}} + N_{x\phi} \frac{\sin\phi_1 \sin\beta}{\ell} + Q_x \frac{\cos\phi_1 \sin\beta \cos\beta}{\ell} + N_{\phi x} \sin\phi_1 \sin\beta \cos\beta + N_\phi \frac{\sin^2\phi_1 \sin^2\beta}{\sqrt{\ell \cos\beta}} + Q_\phi \frac{\sin\phi_1 \cos\phi_1 \sin^2\beta \cos\beta}{\sqrt{\ell \cos\beta}} \qquad (8)$$

The conditions on the other forces and moments $N_{x\phi}^{(1)}$, $Q_x^{(1)}$, $M_{x\phi}^{(1)}$, $M_x^{(1)}$ can be obtained accordingly.

Conditions along the foundation

These ones are displacement conditions (Figure 8). A first relation is :

$$u = \bar{u} \cos\beta - \bar{v} \sin\phi_1 \sin\beta + \bar{w} \cos\phi_1 \sin\beta \qquad (9)$$

Using Vogt's formulae relating the displacements to the forces acting on the same point, we derive (Vogt, 1925) :

$$u = N_x\left(-\frac{k_2}{E_F}\cos\beta\right) + N_{x\phi}\left(\frac{k'_3}{E_F}\sin\phi_1\sin\beta\right) + Q_x\left(\frac{k_3}{E_F}\cos\phi_1\sin\beta\right)$$

$$+ M_x\left(\frac{k_5}{E_F\,\delta}\cos\phi_1\sin\beta\right) \qquad (10)$$

The other conditions related to the displacements v, w and δ_x are written accordingly.

FINITE DIFFERENCE METHOD

The equilibrium and boundary equations are rather sophisticated They use three functions u, v, w with derivatives up to the 4th order. Therefore we chose 6th degree polynomials to integrate them numerically, in order to keep enough accuracy (Figure 9).

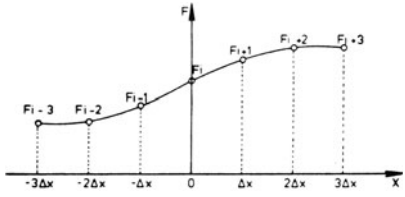


Figure 9. – *6th order polynomial function*

The general form of the derivatives is :

$$\frac{\partial^{m+n} F_{i,j}}{\partial x^m\,\partial y^n} = \frac{1}{\Delta x^m\,\Delta y^n}\,[d(m)]\cdot F_{i,j}\,[d(n)]^T$$

with m+n = 0,1,2,3,4.

The arrays are defined by :

$$d(0) = [\ 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0\ ]$$

$$d(1) = \frac{1}{60}\,[\ -1 \quad 9 \quad -45 \quad 0 \quad 45 \quad -9 \quad 1\ ]$$

$$d(2) = \frac{1}{180}\,[\ 2 \quad -27 \quad 270 \quad -490 \quad 270 \quad -27 \quad 2\ ]$$

$$d(3) = \frac{1}{8}\,[\ 1 \quad -8 \quad 13 \quad 0 \quad -13 \quad 8 \quad -1\ ]$$

$$d(4) = \frac{1}{6}\,[\ -1 \quad 12 \quad -39 \quad 56 \quad -39 \quad 12 \quad -1\ ]$$

And the matrix $F_{i,j}$ by :

$$F_{i,j} = \begin{vmatrix} F_{i-3,j-3} & F_{i-3,j-2} & F_{i-3,j-1} & F_{i-3,j} & F_{i-3,j+3} & F_{i-3,j+2} & F_{i-3,j+3} \\ F_{i-2,j-3} & F_{i-2,j-2} & F_{i-2,j-1} & F_{i-2,j} & F_{i-2,j+1} & F_{i-2,j+2} & F_{i-2,j+3} \\ F_{i-1,j-3} & F_{i-1,j-2} & F_{i-1,j-1} & F_{i-1,j} & F_{i-1,j+1} & F_{i-1,j+2} & F_{i-1,j+3} \\ F_{i,j-3} & F_{i,j-2} & F_{i,j-1} & F_{i,j} & F_{i,j+1} & F_{i,j+2} & F_{i,j+3} \\ F_{i+1,j-3} & F_{i+1,j-2} & F_{i+1,j-1} & F_{i+1,j} & F_{i+1,j+1} & F_{i+1,j+2} & F_{i+1,j+3} \\ F_{i+2,j-3} & F_{i+2,j-2} & F_{i+2,j-1} & F_{i+2,j} & F_{i+2,j+1} & F_{i+2,j+2} & F_{i+2,j+3} \\ F_{i+3,j-3} & F_{i+3,j-2} & F_{i+3,j-1} & F_{i+3,j} & F_{i+3,j+1} & F_{i+3,j+2} & F_{i+3,j+3} \end{vmatrix}$$

STUDY OF THE BUTTRESS

Equilibrium Equations

The study of the buttress is that of a trapezoidal plate loaded in its plane. The equilibrium conditions are derived from Flügge's equations observing that

$$w \equiv Q_x \equiv Q_\phi \equiv M_x \equiv M_\phi \equiv M_{x\phi} \equiv 0 \qquad \text{and } \rho \to \infty$$

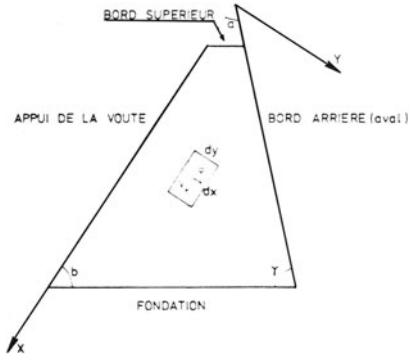


Figure 10. - *Sketch of a buttress*

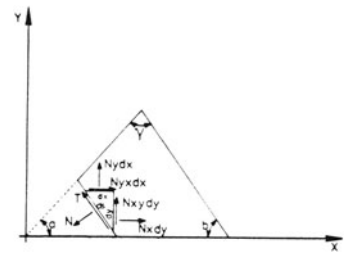


Figure 11. - *Loads acting on the upper edge*

There remain two equilibrium conditions. The first of them has the following form :

$$u'(D')+u'(D\cdot\frac{1-\nu}{2})+u''(D)+u''(D\frac{1-\nu}{2})+v'(D\cdot\frac{1-\nu}{2})+v'(D'\nu)+v''(D\frac{1+\nu}{2})= -\overline{X} \qquad (11)$$

Boundary Conditions

We wrote peculiar conditions for the upper edge (Figure 14), the rear edge, the foundation and the vault linking edge.

But we also had to write specific conditions for the different angle corners, that is the points respectively comprised between : the upper and rear edges, the rear edge and the foundation, the vault and the foundation, the vault and the upper edge.

The finite difference method

We use the method described by Soare, with triangular coordinates, and 2nd degree polynomials integration functions (Soare, 1962).

THE CALCULATION OF THE DAM

Two boundary conditions of the shell along the buttress express displacements, one parallel to the boundary line (Ox) and another perpendicular. The related displacements δ_x and δ_y are known when the buttress has been calculated.
On the other hand, the stress and strain fields in the buttress depend on forces coming from the vault.
The get rid of the indetermination, we proceed the following way: We compute the stress fields in the vault, resulting from unit displacements $\delta_x=1$ (and $\delta_y=1$ respectively) applied at each of the M' nodes on the link edge and considering one non-zero displacement at a time. Thus we obtain 2M' stress fields.
Moreover, we compute the stress field resulting from external loads (e.g. hydrostatic) with zero displacements on the link edge.
On the other hand, we compute the displacement fields in the buttress, resulting from unit loads $\overline{N}_y=1$ (resp. $\overline{N}_x=1$) applied on each node of the boundary, and considering one non-zero displacement at a time, to obtain 2M' displacement fields. Moreover we compute the stress field resulting from the external loads applied to the buttress (e.g. prestressing) and considering zero-loads on the link edge.
Now the solution comes through iteration : we compute the efforts $\overline{N}_y$ and $\overline{N}_{xy}$ resulting from external loads on the load supposed to be fixed. We derive the displacements in the buttress due to them and the external loads, then we introduce the resulting link edge displacements in the vault, and so on.
To keep a good convergence of the iteration it is mandatory to consider interpolated values at each step j and we adopt :

$$\delta^*_j = \frac{3\delta_{j-1} + \delta_j}{4}$$

Finally we stop the procedure when we reach the following condition at each point :

$$\frac{(\delta_{j-1} - \delta_j)_{max}}{\delta_{j_{max}}} \leq 1.10^{-4}$$

Generally, about 20 steps are enough to reach the solution.

STRUCTURE OF THE BAVOMU PROGRAM

The program has an approximate amount of 3000 FORTRAN statements, displayed in 44 functions and subroutines, allowing successive overlay (Figure 12).
The main root contains subroutines defining the basic geometrical and mechanical characteristics of the dam and the algorithm (GOPIV2) used to solve the systems of linear equations (Gauss with pivots-blocks and double precision).

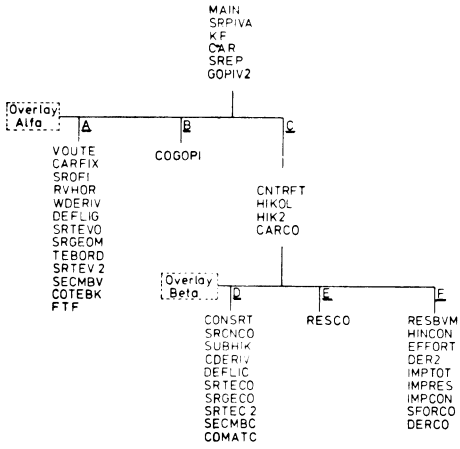


Figure 12. - Tree chart of the routines

The A-branch is devoted to write the coefficients and second members (external loads) of the vault matrix.
The B-branch solves the vault matrix using GOPIV2.
The C-branch is a root for the buttress computation.
The D-branch is devoted to write the coefficients and second members of the buttress-matrix.
The E-branch solves the buttress matrix
The F-branch solves the whole structure where RESBVM performs the iterative procedure using the results of COGOPI and RESCO. Then it computes the efforts of the vault (EFFORT) and of the buttress (SFORCO), and prints the results through IMPTOT and IMPRES.
Thanks to this overlay structure, the CPU size may be limited to less than 256 kbytes. The total mass storage needed is 2Mbytes.
With a 370-158 IBM computer, a dam where we can obtain the values of stresses and displacements in 63 points of half a vault and 63 points of the buttress is calculated in 70 s (printing time excluded).

STUDY OF THE NEBEUR DAM

Characteristics of the dam

This dam was chosen to make comparison with some results found in the Literature (Rochat, 1963 and Falkenberg, 1966). We shall especially refer to the scale model built at the L.N.E.C. in Lisbon.
The main sizes are reported on the Figure 13.

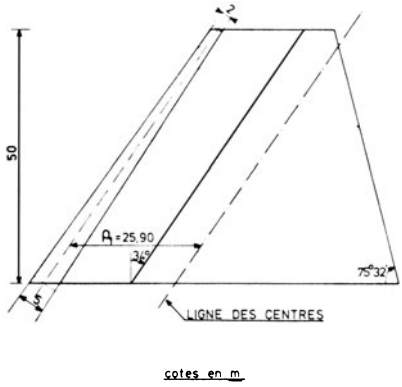


Figure 13. - *Cross section of the studied dam*

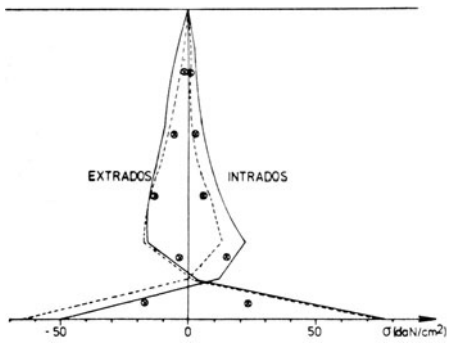


Figure 14. - *Vertical stresses along the crown cantilever (hydrostatic load)*

We successively considered a vault on fixed supports (dotted lines) and on elastic foundations (straight lines).
The Vogt's coefficients relate to ν=0.20 and an equivalent rectangle of 60m by 50m.

Hydrostatic Load

To express it, we just have to write :

$X = Y = 0$ and $Z = -10\,000\, x_1 \cos\beta$

in the equilibrium equations at each node of the vault.
Some significant results are reported on the Figures 14 to 16.

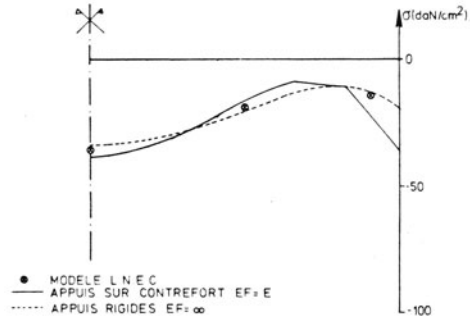


Figure 15. - *Main stresses* (σ_2) *depth : 30m upstream face*

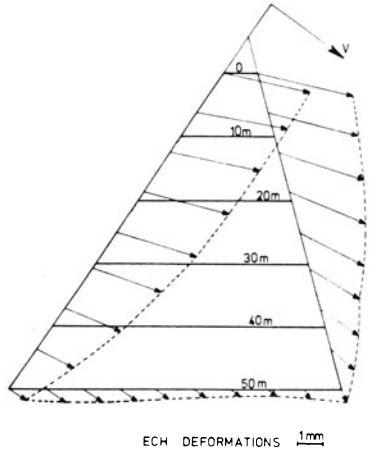


Figure 16. - *Displacements along the edges of the buttress*

Dead load of the vault

To express it, we write the following :

$X = 24\ 000.\delta.\cos\beta$

$Y = 24\ 000.\delta.\sin\beta\ .\sin\phi$

$Z = 24\ 000.\delta.\sin\beta\ .\cos\phi$ (N/m2)

where : ϕ = angular ordinate of the point.
We observed on this fairly raked dam (β=34°) that the weight effect is small and stabilizing.

Thermal effects

We developed a method presented by Priscu (1969) for the study of arch dams. Its interest is that the thermal effect may be introduced by volume and boundary forces, which allows to consider them as external forces.
For instance, the second member of the 3rd equilibrium equation of the shell becomes :

$$VSMB = Z + \alpha \{\frac{1}{\rho} D\theta + \frac{a^2b^2 - 1}{\rho} (K''\theta + 2K'\theta' + K\theta'') - \frac{K}{\rho^2\delta} \Delta\theta$$
$$- \frac{a^2b^2 + 1}{\delta^3} [\Delta\theta(\delta^2K'' - 2\delta\delta'K' + K(2\delta'^2 - \delta\delta')) + \Delta\theta'(2\delta^2K' - 2\delta\delta'K)$$
$$+ \Delta\theta'' (K\delta^2)]\}$$

Using Priscu's procedure, we derived values of θ, θ' and θ'' for each node of the vault and the buttress so that we obtained the diagram of Figure 17 corresponding to an increase of temperature of 15° C along the crest.

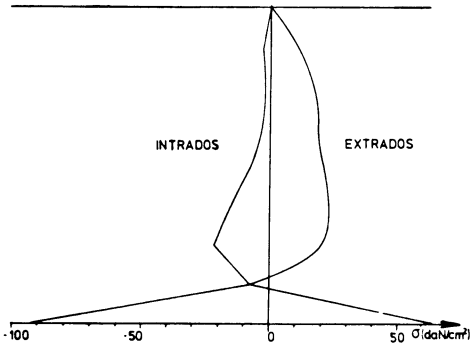


Figure 17. - *Vertical stresses along the crown cantilever (température increase)*

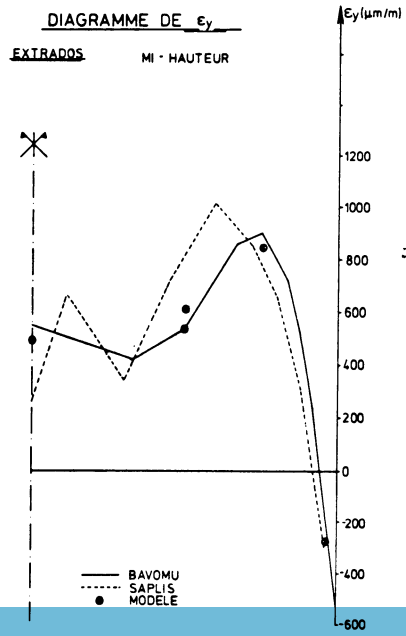


Figure 18. - *Compared results of P.V.C. model, BAVOMU and SAP program*

(upstream face mid-height)

It is worthwile noting that the temperature effect may be as important as the hydrostatic load.

Other effects

It is easy to find that the method is well fitted to consider also other effects such as seismic loads, shrinkage effects and prestressing too.
By the way, considering prestressing and quadratic variation of the thickness, we found very interesting hints to optimize this type of dams.

INTEREST OF BAVOMU

Comparison with a finite element program

It was of interest to compare BAVOMU with a finite element program. We did it by taking the opportunity to use SAPLI5 which is a local improvement of SAP at Liège. SAP uses the method of displacements and isoparametric elements.
On the average the results were almost the same except some local discrepancies shown by SAP along some lines and the boundaries. Therefore we built a P.V.C. model of a clamped vault (5 mm thick, 500 mm high, 256 mm radius, 34° rake) loaded with mercury.
A significant comparison diagram is shown on the Figure 18, and points out the need to check the mathematical models whenever it is possible.
Moreover, it is worthwhile noting that BAVOMU computed the displacements, stresses and efforts in 63 nodes. It needed 236 kbytes and 75 s to compute and print 4700 lines of results.
The SAP program considered 100 nodal points, or 80 elements. The computer used 448 kbytes for 102 s and printed 2400 lines of results. And we do not take account of the time needed to punch the data which is much more important!

Conclusion

BAVOMU is a comprehensive method that allows to study every load acting on a multiple arch dam : hydrostatic load, dead load, seismic effect, ice pressure, thermal variations, shrinkage effect, prestressing on the vault and on the buttress.
The shell may show any cylindrical shape with variable thickness.
BAVOMU allows to have a good knowledge of the actual linkage efforts between the vault and the buttress, which are very sensitive to the elasticity of the structure and the foundation.
Finally, the data are easy to enter, the results easy to interpret and the program may be implemented on any 16 bit mini-computer.

REFERENCES

Arnould, R, (1969) Le calcul des barrages à voûtes multiples par la méthode aux différences finies.
Dr. Thesis, Inst. Génie Civil, University of Liege

Falkenberg, J.C., (1966) An Analytical Investigation into the Structural Behaviour of Cylindrical Shell Buttress Dams.
Southampton University.

Flügge, W., (1960) Statique et dynamique des coques.
Ed. Eyrolles, Paris

Priscu, R., (1969) Contributii la calculul thermoelastic al barajelor arcuite.
Inst. recherches hydrotechniques, nr 14, Bucuresti.

Rochat, M., (1963) Note on the Study of Large Span Multiple Arch Dams.
Symposium on Concrete Models of Dams, Lisbon.

Rydzewski, J.R., (1965) Theory of Arch Dams.
Pergamon Press

Soare, M., (1962) Application des équations aux différences finies au calcul des coques.
Ed. Eyrolles, Paris.

Vogt, F., (1925) Ueber die Berechnung der Fundamentdeformation
Det Norske Videnskaps Akademi, Oslo.

COMPUTER SYSTEM SYNAR FOR NONLINEAR ANALYSIS OF FRAMES AND BEAMS

Michał Saran

Technical University of Warsaw

INTRODUCTION

Most of the contemporary civil engineering codes allow the structuraldesign including plastic reserve of structures. Although theoretical and numerical aspects of the solution of nonlinear problems have been intensively studied, the present state of knowledge is still far from being applied in every-day design. One of the main reasons is the lack of effective computer programs which would enable calculating the structures in the plastic state.

In the paper, the effective methods,algorithms and computer system called SYNAR, which allow analysis for arbitrary frame and beam structures, are presented.

SYNAR makes it possible to calculate the limit load by linear programming (LP - method) and the displacements by quadratic programming (QP - method) in geometrically linear range.

Analysis of large displacements is also possible by employing two methods: the incremental one (QPP - method) and the iterative one (LQP - method).

SYNAR has a convenient input data block and graphical output of the results, which enables automatic plotting of the frame geometry, loading, moments and axial forces charts, deformations of the structure and load-displacement chart.

MATRIX DESCRIPTION OF ELASTIC-PERFECTLY-PLASTIC BEHAVIOUR

Consider an arbitrary structure made from an elastic perfectly-plastic meterial. Let the stru-

cture be discretized, i.e. described by a finite number of parameters that are arranged into the following four vectors: generalized strain $\underline{q} \in R^m$, generalized stress $\underline{s} \in R^m$, generalized displacement $\underline{w} \in R^n$ and generalized load $\underline{p} \in R^n$.
Discretization was made by meaning of the finite element method. The new version of this method was worked out to be used together with mathematical programming.
In this version a new finite element description and connecting operation are used. Details can be found in [1].

Under the assumption of small displacements the strain-displacement relation

$$\underline{q} = \underline{C}\ \underline{w} \tag{1}$$

is linear, with $\underline{C}$ being the / m x n /-matrix of compatibility. The transpose of this matrix enters the linear equilibrium equation

$$\underline{p} = \underline{C}^T \underline{s}. \tag{2}$$

The constitutive relations take the form:

$$\underline{q} = \underline{q}_e + \underline{q}_p,\ \underline{q}_e = \underline{S}^{-1}\ \underline{s},\ \underline{q}_p = \underline{N}^a\ \underline{\lambda}^a,$$

$$\underline{F}^a = \underline{N}^{aT} \underline{s},\ \underline{F}^a \leq \underline{0},\ \underline{\lambda}^a \geq \underline{0},\ \underline{F}^{aT}\ \underline{\lambda}^a = 0, \tag{3}$$

where $\underline{F}^a = \underline{N}^{aT}\ \underline{s} - \underline{c}' = \underline{0}$ are active yield constraints,

$\underline{F}^p = \underline{N}^{pT}\ \underline{s} - \underline{c} < \underline{0}$ passive yield constrains,

$\underline{q}_e$ elastic, $\underline{q}_p$ - plastic strain, $\underline{S}^{-1}$ - elastic stiffness matrix, $\underline{\lambda}$ - plastic multiplier,

$\underline{c} = \begin{bmatrix} \underline{c}' \\ \underline{c} \end{bmatrix}$ - plastic modulus vectors.

LIMIT LOAD ANALYSIS

Employing the rigid-plastic material model the global governing relations discussed in chapter 2 have taken the following form:

$$\dot{\underline{q}} = \underline{C}\ \dot{\underline{w}},\ \underline{p} = \underline{C}^T \underline{s},\ \underline{N}^T \underline{s} \leq \underline{c}, \tag{4}$$

$$\dot{\underline{q}} = \underline{N}\,\dot{\underline{\lambda}}\,, \quad \dot{\underline{\lambda}} \geqslant \underline{0}, \quad \dot{\underline{\lambda}}(\underline{c} - \underline{N}^T\underline{s}) = 0\,.$$

Let $\hat{\underline{p}}$ be a given reference load, e.g. service load. Assuming that loading increases in a proportional manner, i.e.

$$\underline{p} = \mu\,\hat{\underline{p}}, \qquad \mu > 1 \tag{5}$$

one can find an ultimate value μ_u which corresponds to the plastic collapse of the structure. This value μ_u represents a safety factor and can be established by solving the following dual pair of the LP-problems [1, 2]:

$$\text{maximize } \mu$$

$$\text{subject to } \mu\hat{\underline{p}} - \underline{C}^T\underline{s} = \underline{0}, \quad \underline{N}^T\underline{s} \leqslant \underline{c}; \tag{6}$$

$$\text{minimize } \underline{c}^T\dot{\underline{\lambda}}\,,$$

$$\text{subject to } \underline{N}\,\dot{\underline{\lambda}} - \underline{C}\,\dot{\underline{w}} = \underline{0}, \quad \hat{\underline{p}}^T\dot{\underline{w}} = 1, \dot{\underline{\lambda}} \geqslant \underline{0}. \tag{6}$$

On solving, we get the limit load factor, stresses, displacement rates and strain rates at the failure. Using this algorithm, a computer subsystem SYNAR-LP was developed.

INCREMENTAL ELASTO-PLASTIC ANALYSIS

Using equations 1, 2, 3 , the well known set of incremental equations can be obtained [1]:

$$\left[K_i\ (\underline{s}_{i-1})\right]^{ep}\ \Delta\,\underline{w}_i = \Delta\,\underline{p}_i \tag{7}$$

where $\left[K_i\right]^{ep}$ denotes the elastoplastic stiffness matrix, $\underline{s}_{i-1}$ - internal force vector, $\Delta\,\underline{w}_i$ - increment of displacement vector, $\Delta\,\underline{p}_i$ - increment of external load.

In order to solve this problem, the modified tangential stiffness method is used.

Elasto-plastic stiffness matrix for a bar with an active zones formed at the nodal cross-section is derived.

For each load increment a constant stiffness matrix is assumed and calculated on the basis of the internal forces obtained from the previous

load step. The redistribution of the internal forces in all active zones as well as the possibility of unloading are taken into account.

Details can be found in [3].

Using this algorithm, a computer subsystem SYNAR-TS is developed.

ELASTO-PLASTIC ANALYSIS USING QP

It was proved by Maier [4] that the global governing relations of the elasto-plastic structure, using holonomic assumption /the local unloading is excluded/, are equivalent to the following quadratic programming problems:

$$\min \left\{ \frac{1}{2} \underline{s}^T \underline{S} \underline{s} \;\middle\|\; \underline{C}^T \underline{s} = \underline{p},\ \underline{N}^T \underline{s} \leq \underline{0} \right\}$$

$$\max \left\{ -\frac{1}{2} \underline{q}_e^T \underline{S}^{-1} \underline{q}_e - \underline{C}^T \underline{\lambda} + \underline{W}^T \underline{p} \;\middle\|\; \underline{q}_e + \underline{N} \underline{\lambda} - \underline{C} \underline{W} = \underline{0},\ \underline{\lambda} \geq \underline{0} \right\}. \tag{8}$$

Solving it one can find the response of the structure (displacements are of particular interest) to a given load $\underline{p}$ in one step. Details can be found in [5].

Using this algorithm a computer subsystem SYNAR-QP is developed.

FINITE DISPLACEMENTS

By now, displacements of a structure were not taken into account. However, in many cases, for example, in the case of tall frames, displacements must be accounted for.

Its influence is called the $P-\Delta$ effect which gives the additional bending moments due to the vertical loads and finite displacements.

Neglect of the effect can lead to an unsafe design.

P - Δ method [6]

Using the "natural" way at first, a simple method is used. It has clear physical interpretation - the geometric softening is simulated by means of ficticious horizontal forces - but the results depend too much upon a rather arbitrary estimate of the sway.

Two alternative methods called QPP and LQP are here proposed.

Incremental QPP method

From the mathematical point of view, taking into account displacement means the nonlinearity of geometrical equations:

$$\underline{q} = \underline{C}(\underline{w}) \cdot \underline{w} \qquad (9)$$

where compatibility matrix $\underline{C}(w)$ is now variable and depends on an actual field of displacement.

To solve this, the QP problem is employed using incremental steps in the updated Lagrangian formulation.

The basic steps in the incremental procedure are as follows:

1. Assume starting loading level P_i.
2. Perform at the level P_i a geometrically linear deformation analysis by QP to determine deformation field for the frame.
3. Modify compatibility matrix $\underline{C}$ according to the displacement from the previous step.
4. Assume increment of the loading ΔP_i to find current loading level $P_{i+1} = P_i + \Delta P_i$.
5. Repeat steps 2, 3, 4 until the collapse load is reached.

Details can be found in [1].

Using this algorithm a computer subsystem SYNAR-QPP is developed.

Iterative LQP method

Let us now describe a more effective method. It is an iterative method employing Linear and Quadratic Programming. The basic steps are as follows:

1. take an initial configuration as the updated one,
2. solve LP problem in order to find the ultimate load for the updated configuration,
3. solve QP problem in order fo find the displacements of the frame under the load calculated in the previous step,

4. update the configuration by adding the displacements computed in the previous step to the initial nodal coordinates,
5. repeat steps 1, 2, 3 until the difference between subsequent values of the ultimate load factor is sufficiently small.

Details can be found in [1, 7] .

Using this algorithm, a computer subsystem SYNAR-LQP is developed.

COMPUTER SYSTEM SYNAR AND NUMERICAL RESULTS

SYNAR is implemented in FORTRAN on the computer ODRA-1305 /ICL Licence/ under GEORGE 3 system and it is submitted to implement on IBM-360. The package allows analyses of arbitrary planar frames.

The main difficulty in the practical application of presented methods arises from the large dimensions of the matrices $\underline{B}$, $\underline{C}$, and $\underline{N}$. It was therefore necessary to include an efficient generator of these matrices into the program and to organize carefully the interchange of information between the core and the outer memories.
A series of skeleton frames, typical for buildings, were calculated to prove SYNAR accuracy, efficiency and stability.
The first example is of a five-storey frame, calculated by LP method, shown on the left side of Figure 1. On the right hand side of Figure 1 the values are shown of the limit load factors together with collapse mechanisms calculated for various shapes of the yield conditions.
As we can see, the influence of axial forces entering the yield conditions is considerable. In the present example the load factor obtained without axial forces is $\mu_M = 1{,}48$ comparing to $\mu_{M+N} = 0{,}934$ allowing for this influence.

The next example is a skeleton frame of one of Warsaw multistorey buildings (Figure 2). On the left-hand side the geometry and properties of the frame are shown. On the right-hand side a collapse mechanism is drawn.
For the sake of comparison, the four-storey frame shown in Figure 3, was analysed by means of the five presented methods.

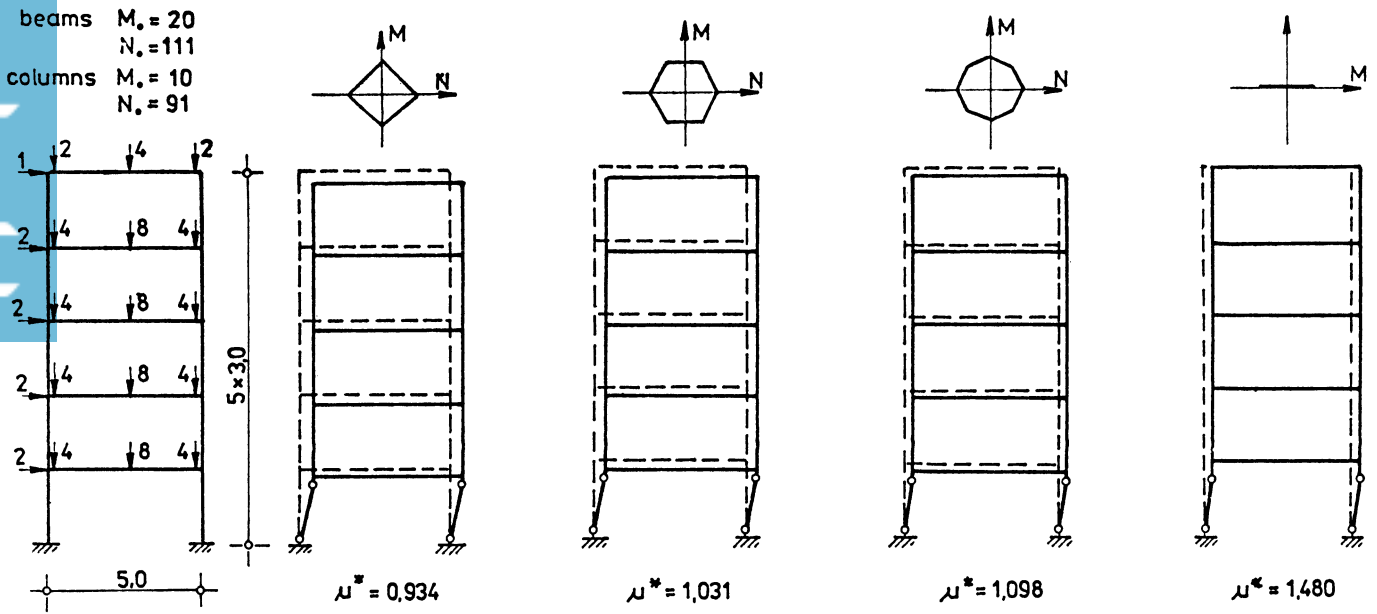
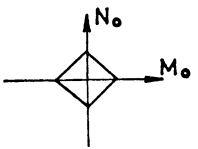

beams M. = 20
N. = 111
columns M. = 10
N. = 91
1
2
4
2
2
4
8
4
2
4
8
4
2
4
8
4
2
4
8
4
5×3,0
5,0
M
N
μ* = 0,934
μ* = 1,031
μ* = 1,098
μ* = 1,480
a/
b/
c/
d/
e/

Figure 1

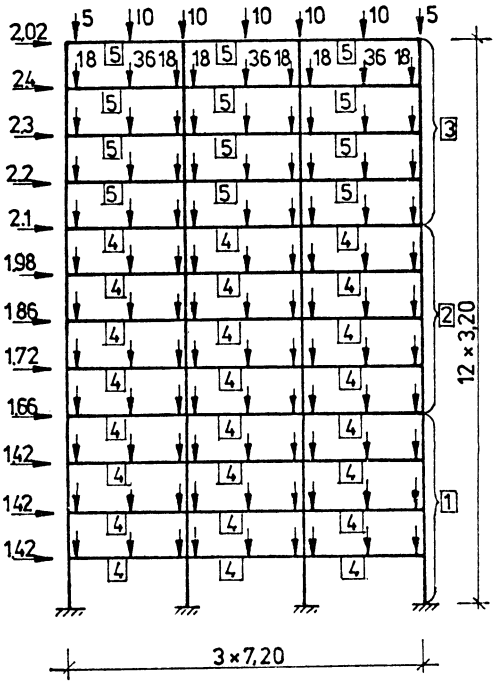
	β	M_o
1	0,1834	271
2	0,1303	148
3	0.1092	63
4	0.1849	98
5	0,1698	86

$$\beta = \frac{M_o}{N_o}$$



[T, m]

$\mu_o = 2{,}0112$



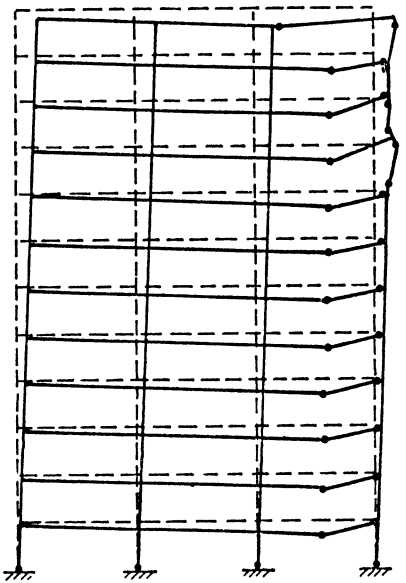


Figure 2

Class no	L. [m]	E [T/m²]	A [m²]	J [m⁴]	M_o [Tm]	N_o [T]
1	7,62	$0,211 \cdot 10^8$	0,00758	$0,599 \cdot 10^{-4}$	22,12	191,87
2	3,048	$0,211 \cdot 10^8$	0,00758	$0,599 \cdot 10^{-4}$	22,12	191,87

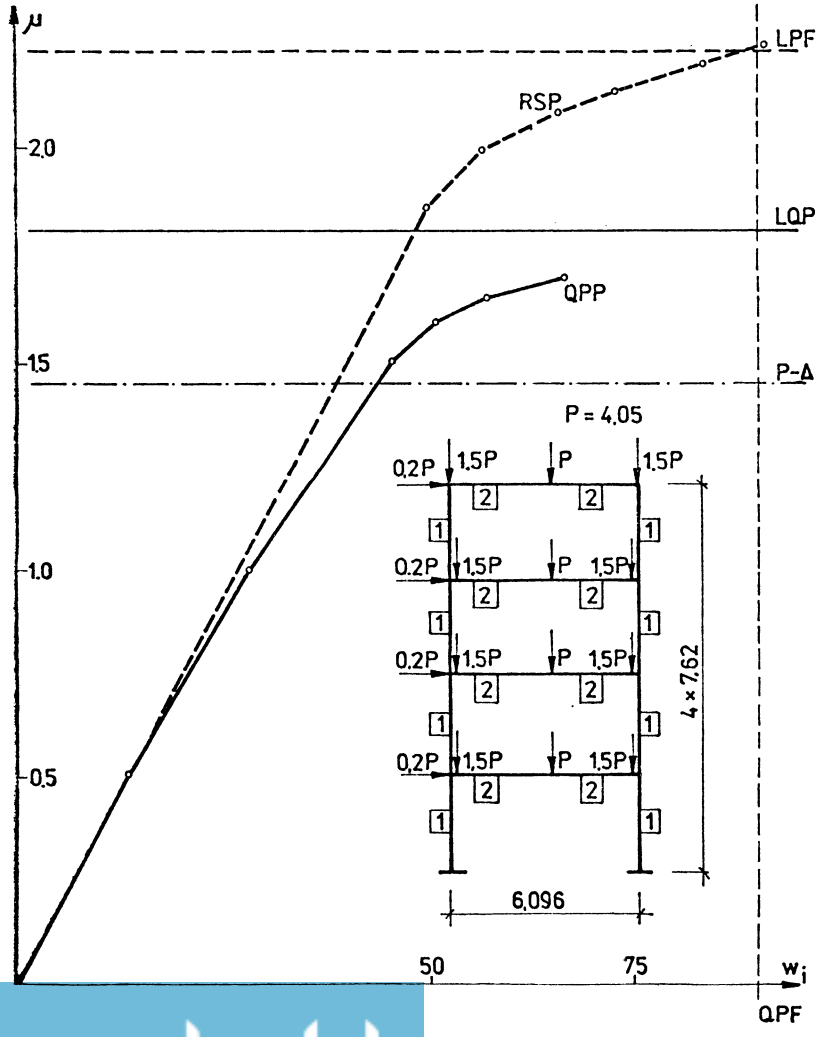


Figure 3

The values of the ultimate load factor and of the sway, obtained by different methods are plotted in Figure 3.
Two geometrically linear models yielded practically the same result: $\mu = 2.238$ by the incremental procedure and $\mu = 2.233$ by the model LP.
The value $\mu = 1.71$ obtained from the updated Langrangian formulation (QPP) must be considered as numerically exact. The LQP - procedure converged after 6 iterations to the value $\mu = 1.81$ which gives an error of about 5%. This is in contradiction to the small displacement analysis that overestimated the load carrying capacity by almost 25%. The P - Δ method showed the underestimation of the ultimate load of the same order / $\mu = 1.46$/.

CONCLUSIONS

Numerical experiments confirmed the fact that steel frames with a small rigidity /for example tall building frames/ should be calculated taking into account axial forces and according to the finite displacement theory.Otherwise unsafe estimates of their load carrying capacity are obtained.

In the case of frames with large rigidities /for example multi-bay, one-storey frames/ the finite displacement influence can be neglected, but the axial forces should be always considered.

System SYNAR is a proper tool for the calculation of planar frames. Ultimate limit state can be computed by the following methods:
LP, TS in geometrically linear range or QPP and LQP with regard to large displacements.

Similarly, for calculations of displacements in the second limit state the methods QP and TS in geometrical linearity or the method QPP with regard to large displacements should be used.

REFERENCES

1. M. Saran, 'Elasto-plastic analysis of frames in small and finite displacement range' /in Polish/, Institute of Fundamental Technological Research Reports, no. 63/1979.
2. M. Saran and A. Borkowski, 'Computer-aided limit analysis of frames' /in Polish/, Archiwum Inżynierii Lądowej /Archives of Civil Engineering/ 24, 645 - 657 /1978/.

3. M. Boni and M. Saran, 'Numerical plastic analysis of plane frames with axial forces taken into account', Archiwum Inżynierii Lądowej /Archives of Civil Engineering/ 24, 629 - 644 /1978/.

4. G. Maier, 'Quadratic programming and theory of elastic-perfectly-plastic structures', Meccanica, 3 265 - 273 /1978/.

5. M. Saran, 'Computer-aided deformation analysis of elastic-plastic frames by quadratic programming', Arch. Inż. Ląd., 28, 1, 1982.

6. M. A. Horne, L. I. Morris, 'Optimum design of multistorey rigid frames', Int. Symp. on Computers in Optimization of Structural Design., University of Wales, January 1972.

7. A. Borkowski, M. Saran, 'On load carrying capacity of frames taking into account finite displacements', Int. J. Comp. Struct., 1982.

ANALYSIS OF PLANE REINFORCED CONCRETE STRUCTURES ACCORDING TO SECOND ORDER THEORY

Ž. Praščević
Civil Engineering Faculty, University of Belgrade
Belgrade, Yugoslavia

INTRODUCTION

It is very often necessary in practice to caary out an analysis and stability control of reinforced concrete structures, especially for large structural systems. Creep and shrinkage of concrete have an important influence on structural behaviour and cannot be neglected in such an analysis. Neglecting of these effects leads to incorect results which are not on the side of safety. Creep and shrinkage are taken into account for meny practical design problems aproximatively. One more accurate procedure, based on the one-dimensional finite element method,is presented in this paper and a computer program is worked out for the determination of internal member forces and joint displacements of the structure.

STRESS-STRAIN RELATIONSHIPS

Reinforced concrete is composed from two materials with different reological properties. We assume steel as an elastic material. Concrete,beside its elasto-plastic instantaneous properties, exibits creep and shrinkage during time. In this paper we assume instantaneous propertis as elastic ones,which is

acceptable in the domain of working stresses under the actual loading in exploatation. Stress-strain relationships in time t are:

- for steel

$$\varepsilon_s(t) = \sigma_s(t)/E_s + \varepsilon_\theta(t) \tag{1}$$

- for concrete, according to the linear creep theory (Aroutynian (1952))

$$\varepsilon_s(t)=\sigma_c(t)E_c(t)+\int_{t_1}^{t} K(t,\tau)\sigma_c(\tau)/E_c(\tau)d\tau+\varepsilon_{sh}(t)+\varepsilon_\theta(t) \tag{2}$$

or

$$\sigma_c(t)/E_c(t)=\varepsilon(t)-\int_{t_1}^{t} R(t,\tau)(\varepsilon_c(\tau)-\varepsilon_{sh}(\tau)-\varepsilon_\theta(\tau))d\tau+\varepsilon_{sh}(t)+\varepsilon_\theta(t) \tag{3}$$

where

$$K(t,\tau) = -E(t)\frac{\partial}{\partial\tau}(1/E_c(\tau) + C(t,\tau)) \tag{4}$$

is the kernel of Volterra's equation (1) and $R(t,\tau)$ its resolvent kernel. Between these kernels exists the next relation

$$K(t,\tau) = R(t,\tau) - \int_{\tau}^{t} R(t,\theta)K(\theta,\tau)d\tau \tag{5}$$

t_1 is the age of concrete when a stress state has appeared, while t is the curent time,

$\varepsilon_s(t)$, $\sigma_s(t)$ and $\varepsilon_c(t)$ and $\sigma_c(t)$ are strains and stresses in steel and concrete respectively,

E_s and $E_c(t)$ are Young's moduli for steel and concrete,

$\varepsilon_{sh}(t)$ and $\varepsilon_o(t)$ are strains due to shrinkage and temperature changes respectively,

$c(t,\tau)$ is the creep measure for the time interval (t,τ)

To simplify the analysis of concrete structures, several authors Ulitskii (1962), Djurić (1963), Trost (1967) and Bažant (1972) have proposed different algebraic stress-strain

relationship for concrete instead of Eqs. (1) and (2). In this paper the algebraic relationship is used in the form proposed by Djuric (1963)

$$E_\theta(t)\varepsilon_c(t) = \sigma_c(t) + (t)\ \sigma_c(t_1) + E_\theta(t)(\varepsilon_{sh}(t)+\varepsilon_\theta(t)) \tag{6}$$

where

$$E_\theta(t) = 2E_c(t)/(2+\phi^*(t,t_1)) \tag{7}$$

$$\rho(t) = \phi^*(t,t_1)/(2+\phi^*(t,t_1)) \tag{8}$$

$\phi(t,t_1)$ is the creep coefficient for the time interval $(t_1 t)$

Eqs. (6), (7) and (8) are the same as the ones proposed by Euro--International Commitee for Concrete (CEB) (1978).

The creep coefficient $\phi(t,t_1)$ and the creep measure are related by the expression proposed by CEB (1978).

$$\phi^*(t,t_1) = E_c(28) \cdot c(t,t_1) \tag{9}$$

STIFFNESS MATRICES AND EQUIVALENT NODAL FORCES FOR A REINFORCED CONCRETE ELEMENT

Let us consider a reinforced concrete element, shown in Fig. 1, exposed to the external forces and temperature changes from time t_1. The strain energy, as a part of the free energy in this thermomechanical process, for the stress-strain relationship (1) is

$$F = \int_0^L \int_{A_c} \Big(\frac{1}{2}\varepsilon_c(t)E_c(t)(\varepsilon_c(t)-2\varepsilon_{sh}(t,t_1)-2\varepsilon_\theta(t)) - \varepsilon_c(t)E_c(t)\int_{t_1}^{t} R(t,\tau)(\varepsilon_c(\tau)-\varepsilon_{sh}(\tau,t_1)-\varepsilon_c^\theta(\tau))d\tau\Big)dA_c dx + \int_0^L \int_{A_s} \frac{1}{2}\varepsilon_s(t)E_s(\varepsilon_s(t) - 2\varepsilon_\theta(t))dA_s\ dx \tag{10}$$

A_c and A_s are the areas of the concrete and steel cross--section respectively, while L is the length of the member.

For the algebraic stress-strain relationship (5), the expression for the strain energy is

$$F^* = \int_0^L \int_{A_c} \left(\frac{1}{2}\varepsilon_c(t)E_c(t)(\varepsilon_c(t)-2\varepsilon_{sh}(t,t_1)-2\varepsilon_\theta(t)) - (t,t_1)\varepsilon_c(t)(E_c(t_1)(\varepsilon_c(t_1)-\varepsilon_\theta(t_1))-E_c(t)(\varepsilon_\theta(t)+\varepsilon_{sh}(t,t_1))\right)dA_c dx$$

$$+ \int_0^L \int_{A_s} \frac{1}{2}\varepsilon_s(t)E_s(\varepsilon_s(t) - 2\varepsilon_\theta(t))dA_s dx \qquad (11)$$

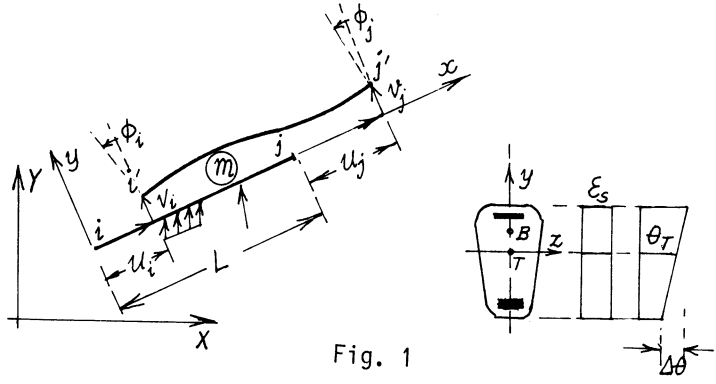


Fig. 1

The concrete and steel strains ε_c and ε_s at some point B (Fig. 1), can be expressed in function of the displacement components in the local member coordinate system (x,y)

$$\varepsilon(t) = \varepsilon^{(l)}(t) + \varepsilon^{(n)}(t)$$

$$\varepsilon^{(l)}(t) = \frac{\partial u(t)}{\partial x} - y\frac{\partial v(t)}{\partial x^2} \qquad (12)$$

$$\varepsilon^{(n)}(t) = 1/2\left(\frac{\partial v(t)}{\partial x}\right)^2$$

where $\varepsilon^{(l)}$ is a linear and $\varepsilon^{(n)}$ is a nonlinear part of the strain ε. The nonlinear term $\varepsilon^{(n)}$ is neglected in the first order theory. Substituting Eq. (9) into Eq. (7), neglecting the terms $\varepsilon_c^{(n)}E_c\varepsilon_c^{(n)}$ and $\varepsilon_s^{(n)}E_s\varepsilon_s^{(n)}$, and taking into account Eq. (3), gives the expression for the strain energy

$$F_m(t) = \int_0^L \int_{A_c} \left(\frac{1}{2}\varepsilon_c^{(1)}(t)E_c(t)(\varepsilon_c^{(1)}(t)-2\varepsilon_{sh}(t,t_1)-2\varepsilon_\theta(t)) -\right.$$

$$- \varepsilon_c^{(1)}(t)E_c(t) \int_{t_1}^{t} R(t,\tau)(\varepsilon_c^{(1)}(\tau)-\varepsilon_{sh}(\tau)-\varepsilon_\theta(\tau))d\tau + \qquad (13)$$

$$\left. - \varepsilon_c^{(n)}(t)\ \sigma_c(t))dA_c dx_1 + \int_0^L \int_{A_s} \left(\frac{1}{2}\varepsilon_s^{(1)}(t)E_c(\varepsilon_s^{(1)}(t)-2\varepsilon_\theta(t)) +\right.\right.$$

$$\left.\left. + \varepsilon_s^{(n)}(t)\sigma_s(t))dA_s dx_1\right.\right.$$

In a similar way, substituting Eq.(9) into Eq.(8), obtains the expression

$$F^*(t) = \int_0^L \int_{A_c} \left(\frac{1}{2}\ {}_c^{(1)}(t)E_c\ (t)(\ {}_c^{(1)}(t)-2\ {}_{sn}(t)-2\ {}_\theta(t)\ -\right.$$

$$- \ (t,t_1)\varepsilon_c^{(1)}(t)(E_c(t_1)(\varepsilon_c^{(1)}(t_1)-\varepsilon_\theta(t_1)) - \qquad (14)$$

$$\left. - E_c\ (t)(\varepsilon_\theta(t)+\varepsilon_{sh}(t,t_1))+\varepsilon_c^{(n)}(t)\sigma_c^{(n)}(t))dA_c dx + \right.$$

$$\left. \int_0^L \int_{A_s} \left(\frac{1}{2}\varepsilon_s^{(1)}E_s(\varepsilon_s^{(1)}(t)-\varepsilon_\theta(t))+\varepsilon_s^{(n)}(t)\sigma_s^{(n)}(t)dA_c dx\right.\right.$$

Expressing the displacements (u,v,ϕ) of eny point B, with local coordinates x and y, in function of the end points displacements $(u_i,v_i,\phi_i,u_j,v_j,\phi_j)$ by the Hermitian polinomials

$$u = (1 - \xi)u_i + \xi u_j$$

$$v = (1 - 3\xi^2+2\xi^3)v_i+(3\xi^2-2\xi^3)v_j +$$

$$+ (\xi - 2\xi + \xi^2)\xi l\phi_i + (\xi-1)\xi^2 l\phi_j \qquad \xi = x/L \qquad (15)$$

$$\phi = 6\xi(\xi-1)\cdot v_i/l + 6\xi(1-\xi)v_j/l +$$

$$+ (1-4\xi+3\xi^2)\ \phi_i + \xi(3\xi-2)\ \phi_j$$

and substituting into (12) and then into (13) and (14), we have

$$F_m(t) = \frac{1}{2}\underline{d}^T(t)\,\underline{S}_{c,m}(t)\Big(\underline{d}(t) - 2\int_{t_1}^{t} R(t,\tau)\underline{d}(\tau)d\tau\Big) -$$

$$- \underline{d}^T(t)\Big(\underline{q}_{sh}(t)+\underline{q}_{\theta}(t)- \int_{t_1}^{t} R(t,\tau)(\underline{q}_{sh}(\tau)+\underline{q}_{\theta,c}(\tau))d\tau\Big) + \qquad (16)$$

$$+ \frac{1}{2}\underline{d}^T(t)\underline{S}_{s,m}(t)(d^T(t)-2\underline{q}_{\theta,c}(t)) + \frac{1}{2}\underline{d}^T(t)\underline{S}_N\,\underline{d}(t),$$

$$F_m^* = \frac{1}{2}\underline{d}^T(t)\cdot\underline{S}_{\phi,m}(t)\underline{d}^T(t) - \rho(t,t_1)\,\underline{d}^T(t)(\underline{S}_{c,m}\cdot\underline{d}(t_1)-\underline{q}_{\phi,\theta}(t_1))$$

$$- \underline{d}^T(t)(\underline{q}_{\phi,sh}(t) + \underline{q}_{\phi\theta,c}(t)) + \frac{1}{2}\,\underline{d}^T(t)\underline{S}_{s,m}(\underline{d}(t)-\underline{q}_{\theta,s}(t)) \qquad (17)$$

$$+ \frac{1}{2}\,\underline{d}^T(t)\underline{S}_N d(t)$$

In these expressions $\underline{S}_{c,m}$, $\underline{S}_{\phi,m}$ and $\underline{S}_{s,m}$ are the member stiffness matrices for concrete and steel respectively, while $\underline{q}_{\theta,c}$, $\underline{q}_{\theta,s}$ and $\underline{q}_{sh}$ are equivalent nodal load vectors due to temperature changes and shrinkage of concrete. These matrices and vectors have the similar form as for an elastic member.

Supposing linear changes for the temperature and shrinkage in a cross-section (Fig. 1)

$$\theta(y,z;t) = \theta_T(t) + \Delta\theta(t)y/h, \quad \varepsilon_s(y,z;t) = \varepsilon_s(t) \qquad (18)$$

$$\underline{q}^T_{\theta,c(s)}=(-A_{c(s)}\theta_T,0,-\Delta\theta I_{c(s)}/h,A_{c(s)}\theta_T,0,\Delta\theta I_{c(s)}/h)\alpha_\theta$$

$$q^T_{sh} =(-A_c\varepsilon_s,0,\ S_c\varepsilon_s,\ \ A_c\ \varepsilon_s,\ 0,-S_c\ \varepsilon_s) \qquad (19)$$

$S_{c(s)}$ and $I_{c(s)}$ are the first and second order moments of the concrete (or steel) area $A_{c(s)}$ for the axis z (Fig. 1).

The matrix S_ϕ and vectors $\underline{q}_{\phi,\theta}$ and $\underline{q}_{\phi,sh}$ can be formulated substituting E_c by $E_{c\phi}$.

Let $\underline{P}(t)$ be the vector of an external force acting on some member point in time t_1, written in the transponed form

$$P^T(t) = (P_x(t),\ P_y(t),\ m(t))$$

then the work done by these forces on the displacements $\underline{u}(t)$

$$\underline{u}^T = (u(t),\ v(t),\ \phi(t))$$

for the member m is

$$W_m(t) = \int_0^L (\underline{u}^T(t_1)\underline{P}(t_1) + \int_{t_1}^t \dot{\underline{u}}^T(\tau)\underline{P}(\tau)d\tau)dx$$

or after partial integration in time τ

$$W_m(t) = \int_0^L (\underline{u}^T(t)\underline{F}(t) - \int_{t_1}^t \underline{u}^T(\tau)\dot{\underline{P}}(\tau)d\tau)dx$$

Substituting Eqs (12) into this expression and integrating along x, we have

$$W_m(t) = \underline{d}_m^T(t)\cdot\underline{q}_F(t) - \int_{t_1}^t \underline{d}_m^T(\tau)\dot{\underline{q}}_F(\tau)d\tau \qquad (20)$$

where $\underline{q}_p$ denotes the nodal forces vector which is equivalent to the external forces. The components of this equivalent vector may be calculated in the same and familiar way as for an elastic member.

Knowing the strain energy $F_m(t)$ and the work of the external forces $W_m(t)$ we can construct the next functional for the member m

$$\Pi_m(t) = F_m(t) - W_m(t)$$

which coresponds to the potential energy for an elastic member.

To formulate this functional for the general structural system, formed from line member elements, we have to transform the member joint forces and displacements from the local (x,y) to the global coordinate system (X,Y) and take into account the structural system topology. This procedure is the same as for an elastic system and will not be repeated here. The functional

$\Pi(t)$ for the whole structural system according to Eqs.(16) and (20) for the integral stress-strain relationship(3) is after, coordinate transformation

$$\Pi(t) = \frac{1}{2}\underline{D}^T(t)(\underline{S}_c(t)+\underline{S}_s+\underline{S}_N)\underline{D}-\underline{D}^T(t)\underline{S}_c(t)\int_{t_1}^{t} R(t,\tau)\underline{D}(\tau)d\tau+\frac{1}{2}$$

$$-\underline{D}^T(t)\underline{Q}_p(t)-\int_{t_1}^{t}\underline{D}^T(\tau)\dot{\underline{Q}}_p(\tau)d\tau-\underline{D}^T(t)(\underline{Q}_\theta(t)-\int_{t_1}^{t} R(t,\tau)\underline{Q}_\theta(\tau)d\tau) \qquad (21)$$

$$-\underline{D}^T(t)(\underline{Q}_{sh}(t)-\int_{t_1}^{t} R(t,\tau)\underline{Q}_{sh}(\tau)d\tau)$$

$\underline{D}$ is the displacement vector of all nodal points,

$\underline{S}_c$, $\underline{S}_s$ are the stiffness matrices of the whole structure for concrete and steel respectively,

$\underline{S}_N$ is the geometric stiffness matrix of the structure,

$\underline{Q}_p$, $\underline{Q}_\theta$ and $\underline{Q}_{sh}$ are the vectors of the equivalent nodal forces due to external loads, temperature changes and concrete shrinkage respectively. The formulation of these matrices and vectors is the same as for an elastic structure (see for example Livesley (1975), Ghali and Neville (1978)).

For the algebraic relationships (6) we have

$$\Pi^*(t) = \frac{1}{2}\underline{D}^T(t)(\underline{S}_{c\phi}(t)+\underline{S}_s+\underline{S}_N)\underline{D} - \rho(t)\underline{D}^T(t)\underline{S}_c(t_1)\underline{D}(t_1) \qquad (22)$$

$$-\underline{D}^T(t)\underline{Q}_p(t)-\int_{t_1}^{t}\underline{D}^T(\tau)\dot{\underline{Q}}_p(\tau)d\tau - D^\tau(t)(\underline{Q}_\theta(t)-\rho(t)\underline{Q}_\theta(t_1)+\underline{Q}_{sh}(t))$$

EQULIBRIUM EQUATIONS AND A COMPUTER PROGRAM

According to the variational principle for a visco-elastic media (Gurtin (1963), Robotnov (1977)), the structural system is in an equilibrium state, at time t, with prescribed boundary conditions (prescribed displacements) when the first variation of the functional $\Pi(t)$ or $\Pi^*(t)$ vanishes (Praščević (1979) i.e.

$$\delta\Pi_{\underline{D}}(t) = 0, \qquad \delta\Pi^{*}_{\underline{D}}(t) = 0,$$

so that we have

$$(\underline{S}_c(t)+\underline{S}_s+\underline{S}_N)\underline{D}(t)-\underline{S}_c(t)\int_{t_1}^{t} R(t,\tau)\underline{D}(\tau)d\tau=\underline{Q}_p(t)+\underline{Q}_{\theta c}(t) - \int_{t_1}^{t} R(t,\tau)\underline{Q}_{\theta}(\tau)d\tau + \underline{Q}_{sh}(t) - \int_{t_1}^{t} R(t,\tau)\underline{Q}_{sh}(\tau)d\tau \tag{23}$$

and

$$(\underline{S}_{c\theta}(t)+S_c+S_N)\underline{D} = \rho(t)\underline{S}_c(t)\underline{D}(t_1)+\underline{Q}_p(t)+\underline{Q}_{\theta}(t)+\underline{Q}_{sh}(t) \tag{24}$$

Some of the displacements $\underline{D}(t)$ are prescribed and these equations may be modified in the same way as for an elastic system. Applying the inverse Volterra´s operator on the system of equations (23), these equations may be written in the form

$$(\underline{S}_c(t)+\underline{S}_s+\underline{S}_N)\underline{D}(t)+(\underline{S}_c+\underline{S}_N)\int_{t_1}^{t} K(t,\tau)\underline{D}(\tau)d\tau = = \underline{Q}_p(t) + \int_{t_1}^{t} K(t,\tau)\underline{Q}_p(\tau)d\tau + \underline{Q}_{\theta,c}(t) + \underline{Q}_{sh}(t) \tag{25}$$

This form is more convinient than (23), because it is easier to determine the kernel $K(t,\tau)$, according to the expression (4), than the resolvent kernel (R,τ) from technical codes or experiments. To solve the system of Volterra´s integral equations (25) numerically we have to divide the time interval (t_1, t) in subintervals assuming

$$\int_{t_1}^{t} K(t,\tau)\underline{D}(\tau)d\tau = (\underline{D}(t_i) + D(t_{i+1})) \cdot k(t_i \cdot t_{i+1})$$

where

$$k(t_i,t_{i+1}) = \int_{t_i}^{t_{i+1}} K(t,\tau)d\tau$$

$$(i = 1, 2, \ldots n)$$

In this case, the probelm is solved first for $t=t_1$ (solution for the elastic system), and then for t_2, t_3, ... t_{n+1}. In every step we use the solutions from the preceding steps. This method gives results of high accuracy in comparisson with an exact solution. The system of algebraic equations (24) is much more easier for solving than the system of integral equations (25).

Knowing the displacement vector $\underline{D}(t)$ we can determine the internal member forces $\underline{q}_m^*(t)$ (axial, shear and bending moments) (Fig. 2) after transforming, the member joint displacement from the global (X,Y) to the member local coordinates (x,y).

The internal member forces are for the integral stress-strain relationships (3)

$$\underline{q}_m^*(t) = (\underline{S}_{c,m}(t)+\underline{S}_{s,m})\underline{d}_m(t)-\underline{S}_{c,m}(t)\int_{t_1}^{t} R(t,\tau)\underline{d}_m(\tau)d\tau$$
$$-q_\theta(t)-\underline{q}_{sh}(t) + \int_{t_1}^{t} R(t,\tau)(\underline{q}_\theta(\tau)+\underline{q}_{sh}(\tau)d\tau \qquad (26)$$

and for the algebraic relationships (6)

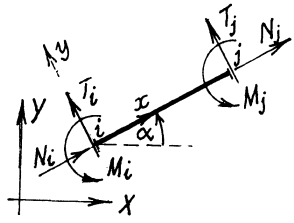


Fig. 2

$$\underline{q}_m^*(t) = (\underline{S}_{\phi,m}(t)+\underline{S}_s)\underline{d}_m(t) -$$
$$-\rho(t)\underline{S}_{c,m}(t_1)\underline{d}_m(t_1)-\underline{q}_{c,\theta}(t)-\underline{q}_{sh}(t)$$
$$-\rho(t_1)\underline{Q}_{c,\theta}(t_1) \qquad (27)$$

where

$$\underline{q}_m^{*T} = (N_i,T_i,M_i,N_j,T_j,M_j)$$

After that we can determine the force reactions at the supported joints.

According to this procedure the program for a computer in FORTRAN programming language is written out. The flow chart diagram of this program for algebraic relationships between stress and strain is given in Fig.3.

The problem solves firstly according to the first order theory for the creep coefficient values $\phi^* = 0$ $(t=t_1)$ and $\phi^* = \phi^*(t)$, $(t > t_1)$. The member axial forces, determined by this theory, are used for calculating of the geomeric matrices, and then the problem is solved by the second order theory.

Input data are: number of joints, number of members, joint coordinates, topology of the structures, prescribed displacements, geometric properties of member cross-sections, materials properties (Young´s moduli for steel and concrete, creep coefficient and shrinkage of concrete), temperature changes and external loads.

Output data are: internal member forces (axial,shear and bending moment at the member ends),force reactions at supported joints and displacements of joints.

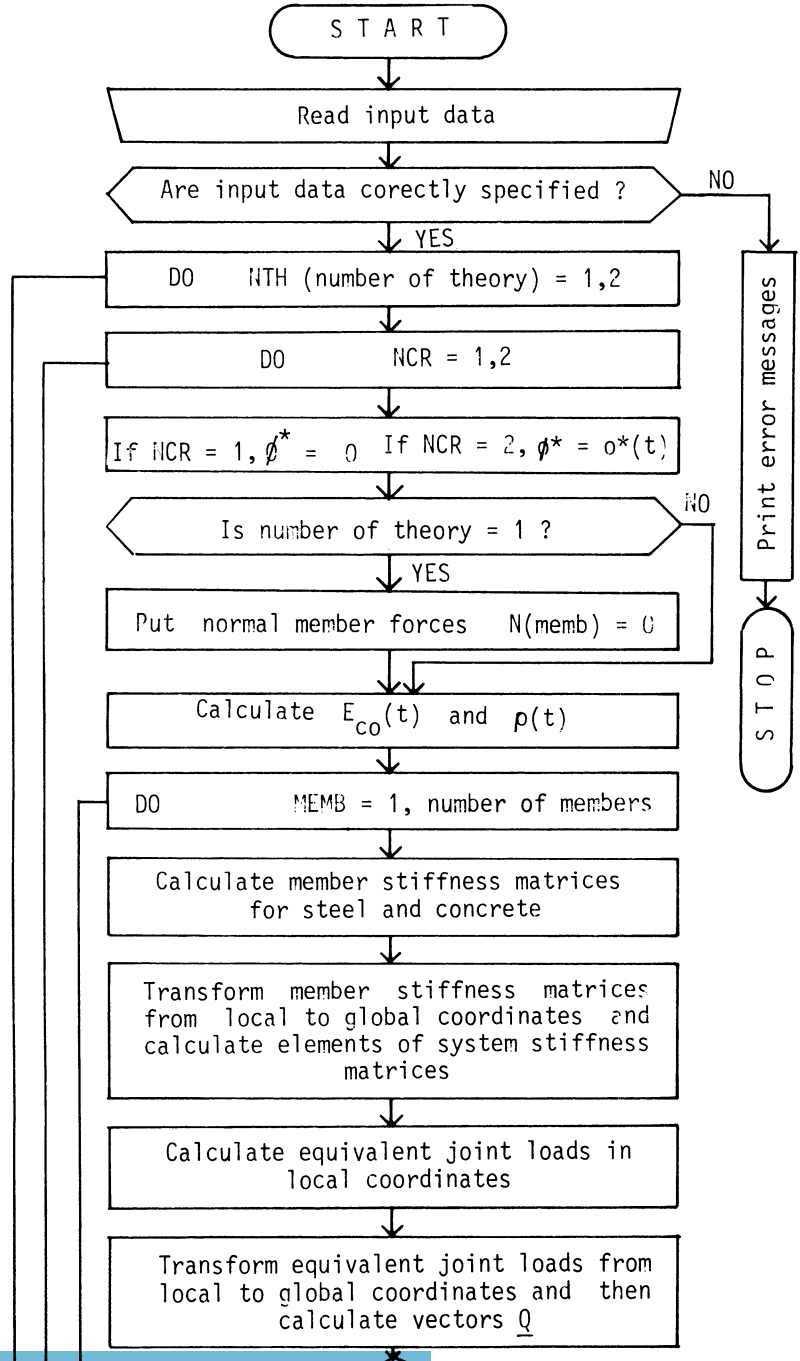


Fig. 3. Flow chart (continued)

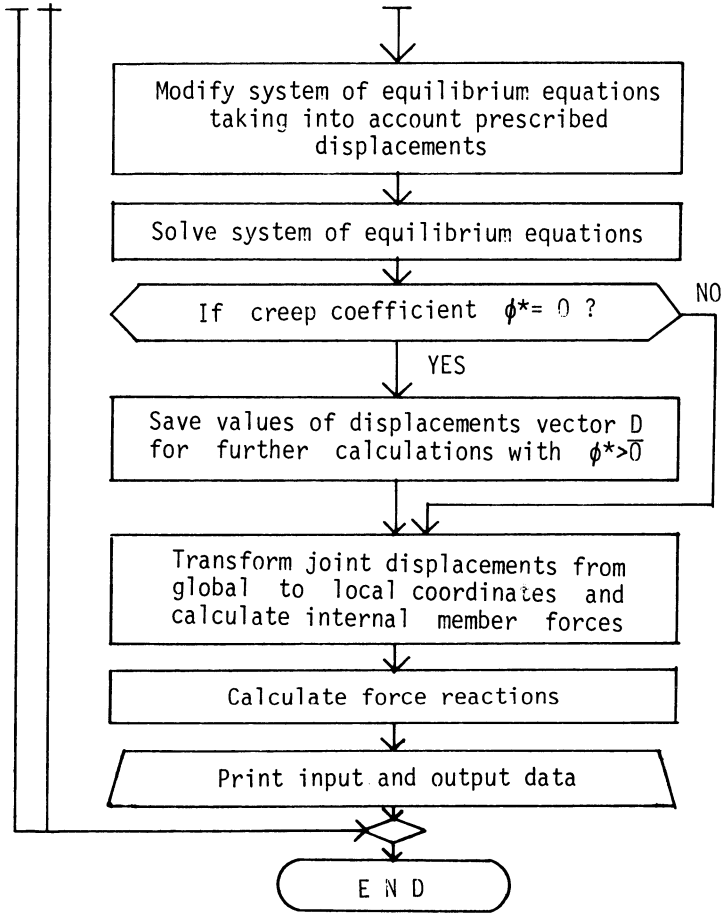


Fig. 3. Flow chart (continuation)

REFERENCES

Aroutyunian, N.Kh. (1952) Some Problems in the Theory of Creep in Concrete Structures (in Russian) Gostehizdat, Moskow

Bazant, Z.P. (1972) Prediction of Concrete Creep Effects Using Age - Adjusted Effective Modulus Method. Journal of ACI, 69, 4: 211-217.

Djurić, M. (1963) Theory of Composite and Presstresed Concrete Structures (in Serbo-Croatian). Naučno delo, Belgrade.

Ghali, A. and Neville, A.M. (1978) Structural Analysis. Chapman and Hall, London.

Gurtin, M.E. (1963) Varitional Principles in the Linear Theory of Viscoelasticity. Archiv for Rational Mechanics and Analysis, 13, 3: 179-191.

Livesley, R.K. (1975) Matrix Methods of Structural Analysis. Pergamon Press. Oxword.

Praščević, Ž. (1979) Nonlinear Theory of Reinforced Concrete Structures. Doctoral Disertacion. University of Belgrade, Civil Engineering Department.

Rabotnov Yu. N. (1977) Elements of Hereditary Theory in Solid Mechanics. Mir, Publ. Moskow.

Trost, H. (1967) Auswirkungen de Superpozitionsprinzips auf Kriech und Relaxationsprobleme bei Beton und Spanbeton. Beton und Stahlbetonbau, 62, 10: 230-238, 11: 261-269.

Ulitckii, I.I., A Method of Computing Creep and Shrinkage Deformation of Concrete for Practical Purposes (in Russian). Beton i Železobeton, Moskow, 4: 174-180.

A TECHNIQUE TO PREDICT THE POST CRACKING BEHAVIOUR OF RCC PLATED STRUCTURES

Shrihari Agrawal

Assistant Professor, Civil Engineering, Maulana Azad College of Technology, Bhopal (India)

ABSTRACT

A technique to predict the Post Cracking behaviour of RCC Plated Structures upto collapse has been described in the paper using Finite Element Incremental Iterative Procedure. Progressive material nonlinearity, cracking/crushing of concrete, yielding of steel are considered in the analysis by making simplified assumptions. Explicit expressions developed by the Author taking into account these factors are used to predict the behaviour. Accuracy and the efficiency of the proposed procedure is established by comparing the predicted response with the experimental results of some of the test reports available in literature.

INTRODUCTION

Prediction of Post Cracking behaviour of RCC Structures is very important due to the present concept of limit state design procedures to be adopted as per the latest codes of practice. The problem becomes sufficiently complex because of non homogenous nature of material, changing topology and properties of structure due to progressive cracking/crushing of concrete and yielding of steel, uncertain material laws and failure criteria. For General loading conditions and typical structures even the elastic analysis becomes difficult and almost there are no methods available to predict the ultimate load. As such it becomes difficult to check/design these structures. Only a powerful method like the finite element method using high speed computers can be successfully employed for solution of such complex structures.

In the past, two types of attempts have been made to analyse such structures using finite element method. The two approaches are the 'modified EI' approach (1) in which cracked elements are assigned properties based on empirical relationships using cracked beam concept and the 'layered element' approach (2,3), in which every element is assumed to be consisting of a number of parallel layers having uniform material properties. The empirical relations used in the 'modified EI' approach are not always realistic. In addition, in both the approaches, the finite element mesh subdivision has got to be sufficiently fine in order to yield reasonably accurate results. This requires large computer storage and considerable manual and computational efforts.

The present paper describes an automatic incremental iterative procedure which can predict deflections, crack patterns and yielding of steel in case of RCC plated structure like Rectangular slabs, folded plates, cylindrical shells, box girders etc. even with a coarse mesh. More realistic element properties are used taking into account nonlinear material properties and degeneration of element stiffnesses after the onset of cracking. Suitability and accuracy of the proposed procedure are illustrated with the help of examples.

FINITE ELEMENT FORMULATION

A flat shell rectangular element obtained by combining a 12 degree of freedom (d.o.f.) membrane element with a 12 d.o.f. plate element is used for the analysis. Such an element has been used (4-6) for the elastic analysis of folded plate type of structures extensively.

With the help of Kirchoff's assumptions, it is possible to find out deformations and strains at any point in the flat element in terms of the deformations of the reference plane

$$\{\epsilon\} = \{\epsilon_0\} + z\{\chi_0\} \qquad (1)$$

where $\{\epsilon\}$ is strain at a point at distance 'z' from reference plane and $\{\epsilon_0\}$ and $\{\chi_0\}$ are strains and curvatures of the reference plane.

The stiffness [K] of such an element can be obtained as follows :

$$[K] = \left[\begin{array}{c|c} [K_m] & [K_{mb}] \\ \hline [K_{bm}] & [K_b] \end{array}\right] \qquad (2)$$

where m, b, mb or bm refer to membrane, bending and membrane-bending interation. The membrane-bending interaction term exist due to unsymmetrical material properties about the reference plane. Such a situation is always present in RCC due to composite nature of the material, cracking and/crushing of concrete and yielding of steel.

Explicit expressions for all possible cases of RCC section have been developed (7) to define the material matrices completely, which are required in the formulation of $[K_m]$, $[K_{mb}]$ and $[K_b]$.

Material properties

A nonlinear stress-strain curve for concrete is used and a simplified maximum strain failure criteria is used to define cracked/crushed depths of concrete. Thus a more realistic assessment of rigidities, forces resisted and crack penetration etc. is made in the proposed analysis.

Numerical integration technique has been used to account for the varied material properties or the release forces within the element e.g.

$$[K]^e = \int [B]^T [D] [B] \, dv$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n} [B_{ij}]^T [D_{ij}] [B_{ij}] C_i C_j ab \qquad (3)$$

where, $[B_{ij}]$ and $[D_{ij}]$ are respectively the strain derivative matrix and the material matrix and C_i and C_j are the weighting constants at the Gauss point (i,j) of the rectangular element having sides a and b.

INCREMENTAL ITERATIVE PROCEDURE

For the analysis the desired load is applied in a number of load steps and the incremental iterative procedure adopted is as described below :

For the first load step and in the first iteration, elastic uncracked properties are used to determine

element stiffness to start with. Structural stiffness is assembled and the incremental nodal deformations are obtained from which the incremental strains at Gause points of every element are obtained. At each gause point, total strains are obtained by adding to the values obtained in the previous iteration, the incremental strains. Similarly total anticipated forces are obtained by adding the incremental forces based upon latest available rigidities of the previous iteration. Actual forces resisted are calculated based on the total strains. Release forces are obtained and unbalanced nodal forces are calculated. Vectorial norm of unbalanced nodal forces is checked against the specific convergence criteria and accordingly either the next load step is applied or iterations are carried out by taking the unbalanced forces as the load vector. If the unbalanced nodal forces are large and divergence is indicated to show the failure, either the analysis is stopped or a search for the ultimate load is made by reducing the size of the last load step. A limit on the iterations in any load step is also specified to avoid the procedure form being too costly.

TEST EXAMPLES

To establish the efficiency and accuracy of the proposed method number of test examples of folded plates, cylindrical shell, box girder etc. were analysed. Two test examples described below are selected for comparing the analytical results obtained by author with the reported experimental results.

The examples concern two models of a folded plate having butterfly type of cross section and of approximately one sixth geometrical scale of a prototype canopy over a school walkway tested by Scordelis and Gerasimenko (8) to study the behaviour of such structures. The two models were of identical geometrical dimensions but differed in the provision of reinforcement as shown on Fig.1. Reinforcement requirement for model 'A' was determined by elementary beam theory. The overall plan dimensions of the models are 762 mm x 1778 mm and the plates are 12.7 mm thick and cast with 38.1 mm thick end diaphragms. The concrete cylinder compressive strength for model 'A' after 16 days is given to be 20.67 N/mm^2 and that for model 'B' at 14 days equal to 25.011 N/mm^2. Poisson's ratio of concrete equals 0.18. In both the models, 14 gauge 2.032 mm

diameter annealed tie wire has been used as steel reinforcement. The yield stress, the ultimate strength and the modulus of elasticity are respectively 227.37 N/mm^2, 358.28 N/mm^2 and 196.365 KN/mm^2. Line loads equal to the joint reactions produced by a uniform design load of 2.7512 KN/m^2 were applied at a number of points on the fold lines as indicated in Fig. 2(a) Vertical displacements of the joints at mid span and the strains in the longitudinal reinforcements were observed in the two models on 15 cm gauge length and these are shown in Fig. 3 and Fig. 4, for folded plates A and B respectively.

PROPOSED THEORETICAL ANALYSIS

A quarter of the folded plate is discretized as shown in Fig. 1 into eight elements and fifteen nodes with appropriate boundary conditions. The actual reinforcement as shown in Fig.1 is idealized into four types of steels with bars assumed to be placed at middle of the thickness. A total load equal to 4.5 times the design load is applied in five steps, the first four steps being equal to the design load. The maximum of five iterations are carried out in each of the first four steps. In the fifth load step, analysis is terminated after first iteration due to divergence in the residual load. The total time required for the analysis is 11 min. 56 secs. and 12 min. 34 secs. for models A and B respectively on IBM 360/44 computer.

Comparison of results

The longitidunal stresses and transverse moments at midspan under design load as obtained by using folded plate theory have been reported. These are compared with the results of the proposed procedure in Fig.2. The small differences between the two are partly due to coarse discretization adopted and partially due to consideration of different material properties in different regions on account of variation in reinforcement in the proposed finite element analysis. The classical folded plate theory assumes uniform material properties throughout. The deflections at different points are compared in Figs.3 and 4. The predicted deflections are more than the actual deflections because the elements in the valley are found to be completely cracked at design load turning the structure to be very flexible. A similar trend is observed in predicted strains due to the

above reason and partly due to the fact that the observed strains are the average strains on 15 cm gauge length. The results would obviously improve considerably by using finer mesh in the valley. The crack patterns are shown in Fig.5 which are almost similar for both the models. They compare well with the reported crack patterns as observed in actual test. The failure load is predicted 'exactly' as after first iteration in the fifth load step, the program gets terminated due to non-convergence.

CONCLUSIONS

The proposed procedure which uses a simple material idealisation is capable of simulating the dominant features like cracking, crack penetration and yielding of steel in RCC plated structures for general loading condition.

The method is more realistic than the empirical modified EI approach and more economical than the layered approach. It can be adopted on medium size computers like IBM 360/44.

Explicit expressions for membrane, bending and interaction behaviour greatly facilitate the procedure specially when used alongwith numerical integration technique.

ACKNOWLEDGEMENT

The Author is thankful to Principal, Maulana Azad College of Technology, Bhopal and University of Roorkee, Roorkee for providing the necessary help in conducting the above investigation.

REFERENCES

(1) Bell, J.C. (1970) A complete analysis for reinforced concrete slabs and shells, Ph.D. Thesis, Canterbury University, Newzealand.

(2) Hand,F.R., Pecknold,D.A. and Schnorbrich,W.C. (1972) A Layered Finite Element Nonlinear Analysis of R.C.plates and Shells. Structural Research Series No. 389, Deptt. of Civil Engg. University of Illinois.

(3) Lin,C.S. (1973) Nonlinear Analysis of R.C. Slabs and Shells, Structural and Material Research

Report No. UCSESM 73-7, Deptt of Civil Engg., University of California, Berkley.

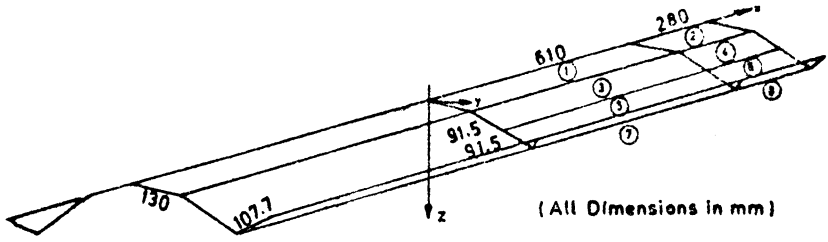
(4) Macleod,I.A. (1969) New Rectangular Finite Element for Shear Wall Analysis, Jr. Str. Div. Proc. Am. Soc. Civil Engrs. Vol. 95, pp 399-409.

(5) Lin,T.K. (1971) Elastic Analysis of Bridge Structures by Finite Element Method, Ph.D. thesis, Imperial College, London.

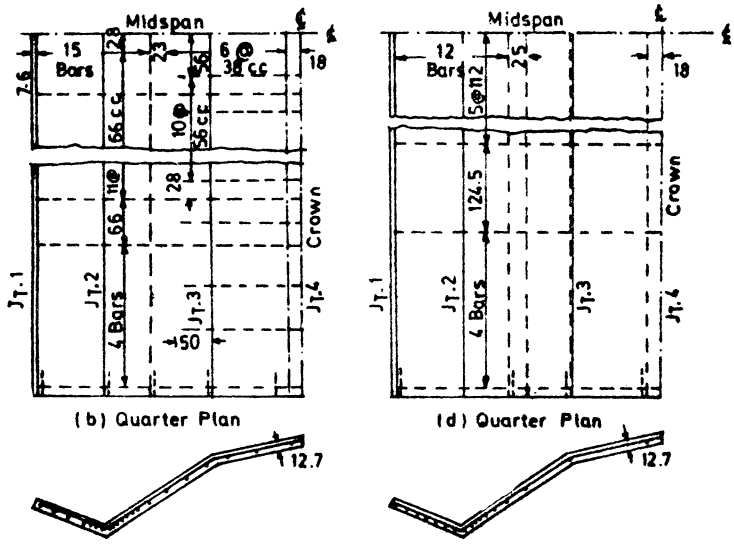
(6) Gibson,J.E. (1964) An Investigation of Folded Plate Structures, The Structural Engineer, Vol.42, No.9, pp 299-304.

(7) Agrawal, Shrihari (1979) Prediction of Post Cracking Behaviour of RCC Plate Structures, Ph.D. thesis, University of Roorkee, Roorkee, India.

(8) Scordelis,A.C. and Gerasimenko,P.V. (1966) Strength of Reinforced Concrete Folded Plate Models, Jr. of Str. Div. Proc. of Am. Soc. of Civil Engrs. Vol.92, No. ST-1, pp 351-363.



(a) Finite Element Idealization of a Quarter



(b) Quarter Plan

(c) Midspan Section

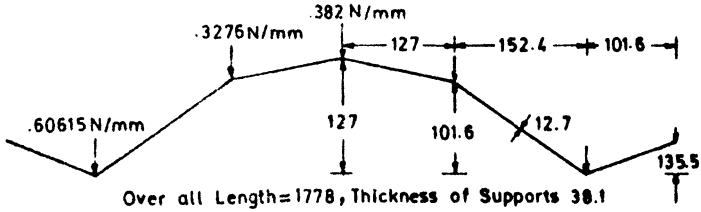
Reinforcement for Model A Based on Folded Plate Method.

(d) Quarter Plan

(e) Midspan Section

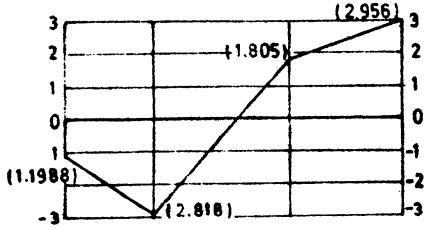
Reinforcement for Model B Based on Beam Theory.

Fig.1: Details of Folded Plates Tested by Scordelis and Gerasimenko.

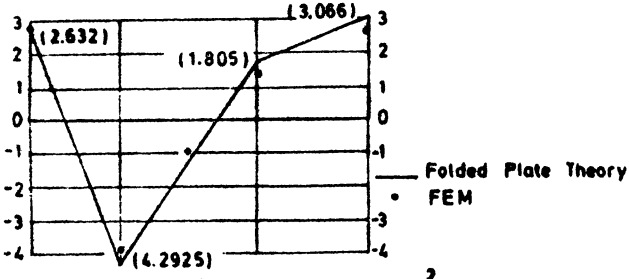


Over all Length=1778, Thickness of Supports 38.1
(All Dimensions in mm)

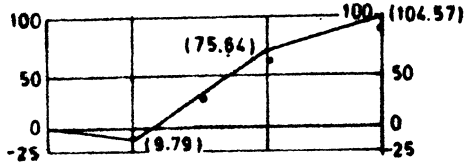
(a) Full Design Line Load and Basic Dimensions of Model A & B



(b) Midspan Longitudinal Stress (N/mm^2) Under Design Load (Beam Theory)

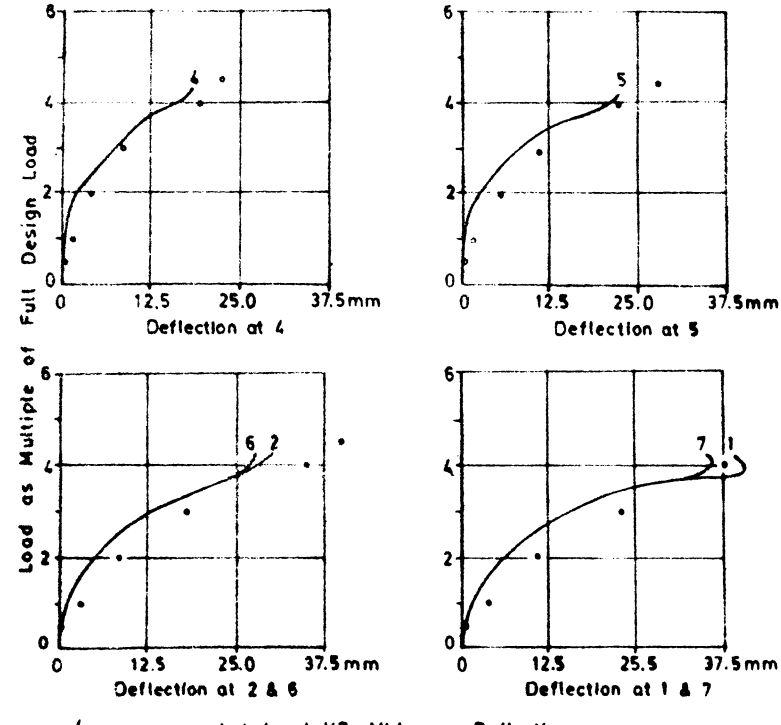


(c) Mid Span Longitudinal Stress (N/mm^2)



(d) Mid Span Transverse Moment (mm N/mm)

Fig 2. Elastic Analysis at Design Load of Folded Plates A & B tested by Scordelis and Gerasmenko.



(a) Load VS Midspan Deflections

(b) Load VS Midspan Longitudinal Strain 10^{-4} (mm/mm)

Fig. 3. Comparison of Results for Folded Plate 'A'

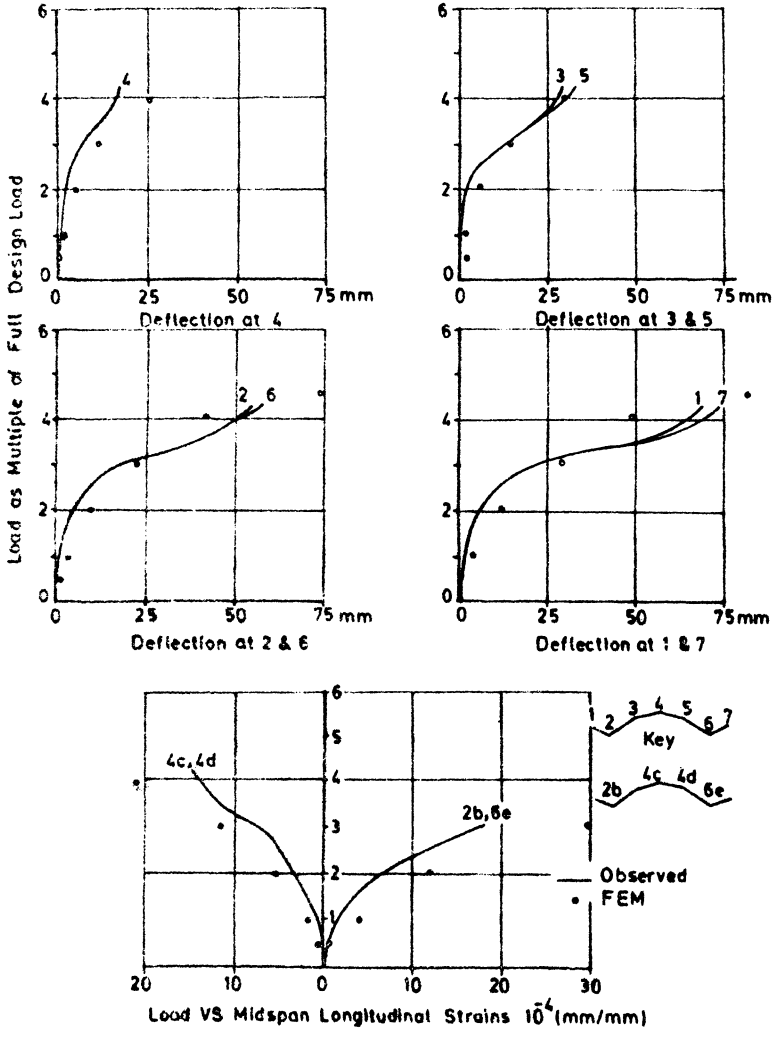


Fig 4. Comparison of Results for Folded Plate 'B'

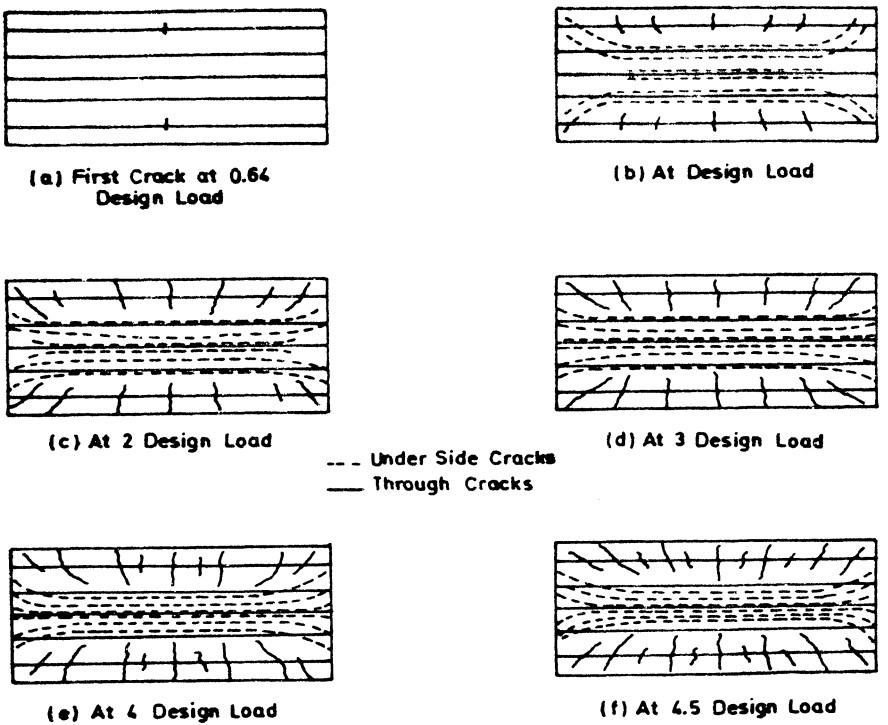


Fig 5. Crack Patterns for Folded Plate 'A'

MECHANICAL ENGINEERING

A GENERAL PROGRAM FOR THE OPTIMUM SYNTHESIS OF PLANE MECHANISMS

R. Avilés; Mª.B. Ajuria; J.A. Tárrago.
Escuela Superior de Ingenieros Industriales.
Bilbao, (SPAIN).

ABSTRACT

In this paper a new method is proposed for the optimum synthesis of plane mechanisms. Coupler-curve, function generation, rigid-body guidance and any combination of these three kinds of synthesis way be solved with this method, which uses the same objetive function for the solution of all these problems. In this work we present a new approach based in minimizing the deformation of the bars which forms the mechanism when it is compulsed to verify geometric constraints along the synthesis positions. The different way in discretizing the mechanism enables us to do all these kinds of synthesis with the same concept. This method is very easy to implement in a digital computer and the resulting programs are simple, general and homogeneous.

INTRODUCTION

The same way it happens with the kinematic and dinamic analysis of mechanisms, the synthesis is approached from very different points of view. Nevertheless in this case the problem is extraordinary more complicated because of the difficult we find even in establishing the initial concept where we can base the synthesis method.

Refering to the optimum synthesis of mechanisms, we can find very few methods; this fact is meaningful because we have not only the synthesis difficultie we have already exposed but also the inherent difficul-

ties of the optimization method. The interest of the optimum synthesis methods is important because from the point of view of the error, are comparable to those of the apro imated synthesis, and even in the case of the exact synthesis the tolerances in manufacture and work conditions introduces real errors of similar or superior order than those produced by the synthesis. The optimum synthesis also permits the implementation of program packages like the MECPLAN from which the one we are presenting here is an integrating part.

Among the different methods for the optimum synthesis of mechanisms we can remark those refering to concrete mechanisms, see references [1] and [2].Other methods, among which we can find the one presented by Klein [3], has its application field limited by a restriction to a kind of synthesis. Anyway, these methods, uses initial concepts with small or null relation with any analysis method we know, and this complicates the packages of designing programs of mechamisms. The main reason for this difference between the concept of analysis methods and the concept of optimum synthesis consists, perhaps, in the objective function used which in a majority of the methods, is based in some kind of geometrical error like this:

$$e_r = \sum_{i=1}^{S} w_i (c_{d_i} - c_i)^2 \qquad (1)$$

S=number of synthesis positions
c=characteristic of the mechanisms; it can be angles between bars, coordinate nodes, etc.
c_d=the desired characteristic
w_i=the ponderation coefficient

It is obvious that the mechanisms which minimize the error defined in equation (1) are optimum ones according to this assessment. If the assessment adopted had been different, would have obtained a different solution, but, anyhow optimum according to this assessment. The validity of an optimum synthesis method is so determined by the correctness of the chosen objective function.

In equation (1) we can see the necessity of obtaining function c_d for each kind of synthesis and each kind of mechanism. This is a hard and complicate

work when the number of kinematic configurations we want to take into account is high, even more if we take into account that within each configuration the expressions of the c_d are different for each node, element, and kind of synthesis. All this is in favour of the opinion that mantains the little generality we can obtain using a function like (1).

The method we are proposing in this paper uses a function that is not based in minimizing a geometrical error, which convergence to a solution is guaranteed even when the characteristic generated by the initial mechanism is far away from the one we want (see the examples).

OBJECTIVE FUNCTION

Let´s have a discretized mechanism (ref.[4]) using b bars, which lengths can be grouped in a vector of length {L}. We are going to suppose, at first, that the positions of the nodes joined to the fixed element (we will demostrate later on that this condition is not necessary) are also data of the problem This mechanism, defined this way, is the initial point for the synthesis.

If we break the condition of rigid body of the composing bars, we can make the mechanism satisfy any geometrical (and also kinematic) restriction c. In general condition c will not belong to the field of possible rigid body motions of the mechanism, and the problem of obtaining the equilibrium position for each condition c consists in solving the deformated position problem in reference [4].

Taking into account that we must obtain the deformate position for each of the S synthesis positions we can think that a valid optimization synthesis can be the one of obtaining the mechanism which deformation is minimum in order to satisfy the S synthesis conditions. The way of obtaining the deformated position consists in looking for the lengths {L} that minimizes the function:

$$\varphi = \sum_{i=1}^{b} (l_i - L_i)^2 \qquad (2)$$

.See reference [4].

According to what we have described previously

the objective function that we propose is:

$$\Phi(\{L\}) = \sum_{j=1}^{S} \varphi_j = \sum_{j=1}^{S} \left[\sum_{i=1}^{b} (l_{ij} - L_i)^2 \right] \quad (3)$$

and if we want to use ponderation coefficients:

$$\Phi(\{L\}) = \sum_{i=1}^{S} w_j \sum_{i=1}^{b} (l_{ij} - L_i)^2 \quad (4)$$

The optimum mechanism for a certain configuration will be the one that makes minimum $\Phi(\{L\})$ defined by equation (4), we are going to use a quasi-Newton method similar to the one used in reference [4].

Developing function $\Phi(\{L\})$ in Taylor's series until quadratic terms, round point q that is the q-th aproximation to the minimun.

$$\Phi(\{L\}) \simeq \bar{\Phi}_q(\{L\}) = \Phi(\{L\})_q + \{\nabla\Phi\}_q^T(\{L\} - \{L\}_q) + \\ + \tfrac{1}{2}(\{L\} - \{L\}_q)^T [J_s]_q (\{L\} - \{L\}_q) \quad (5)$$

$[J_s]_q$ is the matrix of second derivatives of Φ evaluated in the point $\{L\}_q$. The minimum would be obtained for:

$$\{\nabla\bar{\Phi}\}_q = \{\nabla\Phi\}_q + [J_s]_q (\{L\} - \{L\}_q) = \quad (6)$$

that can also be written:

$$[J_s]_q \{L\} = [J_s]_q \{L\}_q - \{\nabla\Phi\}_q \quad (7)$$

being

$$d_{ij} = l_{ij} - L_i \quad (8)$$

It results:

$$\frac{\delta\Phi}{\delta L_k} = 2 \sum_{j=1}^{S} w_j \sum_{i=1}^{b} d_{ij} \frac{\delta d_{ij}}{\delta L_k} \quad (9)$$

At this moment, we must obtain the value of $\delta d_{ij}/\delta L_k$, and for this, we need to obtain the relation between the length of the bar $\underline{i}$, when the c_j characteristic is generated, and the initial length of the bar $\underline{k}$. In this work we have made an approximation consisting in suppose that the deformation of

one of the bars does not depend on the length of the others. Obviously this approximation causes a certain amount of error in the method, but it is very useful especially because the remarkable simplification resulting on the search of the solution, and in the computer programs. With this approximation, we have that:

$$\{\nabla\Phi\} = -2 \sum_{j=1}^{S} w_j (\{l\}_j - \{L\}) \qquad (10)$$

and also:

$$[J_s] = 2 \sum_{j=1}^{S} w_j [I] \qquad (11)$$

where [I] is the unitary matrix.

Replacing the equations (10) and (11) in the system (7), and so calculating the value of {L} we obtain:

$$\{L\} = \frac{\sum_{j=1}^{S} w_j \{l\}_j}{\sum_{j=1}^{S} w_j} \qquad (12)$$

The simplicity of the expression (12) is due without any doubt, to the approximation we have made, and shows that having an efficient method to obtain the lengths {l} in each j position it is very easy to obtain the value of the new lengths $\{L\}_{q+1}$, that we expect to find nearer the minimum than $\{L\}_q$ from the previous iteration.

Refering to the convergence of the described method, it has been verified, through many examples that the function decreases quickly (see fig. 1) during the first iterations and then, little by little it diminzes the decrease speed. The end of the minimization is reached when the value of the function Φ is small enough or when the difference between the value of Φ in two consecutive iterations becomes minor than a certain value ϵ, small enough.

In figure 1 we present the results obtained from the minimization of Φ for different mechanisms and kinds of synthesis.

Next we are going to describe the way we make the different kinds of synthesis we have mentioned before with the ideas developped here.

COUPLER-CURVE SYNTHESIS

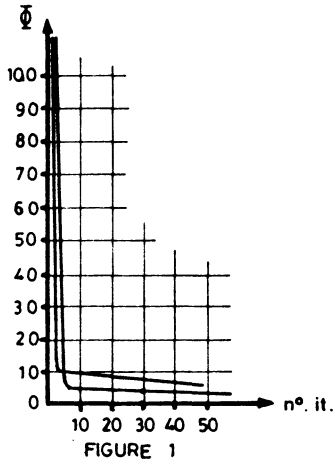


FIGURE 1

Coupler-curve synthesis is, may be, the most studied and the one people have devoted maore effort to. It can be defined as the synthesis dealing with the study of the correspondance between the trajectories described by points of a mechanism and others we have previously specified. It is obvious that the numeric treatment of this kind of synthesis must be done through the study of a series of points over the trajectory (see figure 2).

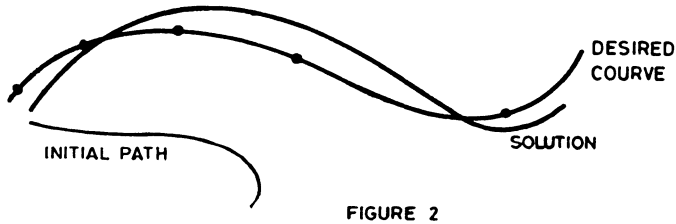


FIGURE 2

Coupler-curve synthesis may include restrictions in the initial parameter of the mechanism, it is: we want the value of the input parameter to be a determined one when the point of the coupler is in a certain position. Using the method we are proposing, we can also include the case of simultaneous generation of courves by several points of a mechanism.

In the synthesis without restrictions on the input parameter, the data we need to define are only the coordinates (x_j, y_j) of the S points of the trajectory. In the event of being restriction, the data will be (x_j, y_j, θ_j), being θ_j the value of the input parameter in the j synthesis position. If we also want to use ponderation coefficients, we will need to use four data: $(x_j, y_j, \theta_j, w_j)$; w_j coefficient only has effect on the equation (12), being the rest of the process inde-

pendent of these values.

Figure 3 represents a discretizated mechanism to achieve a coupler-courve synthesis (coupler point C) without constraints in the input parameter. In this case, the deformated position problem (ref.[4]) should be solved imposing the condition of the node C position being the one we want.

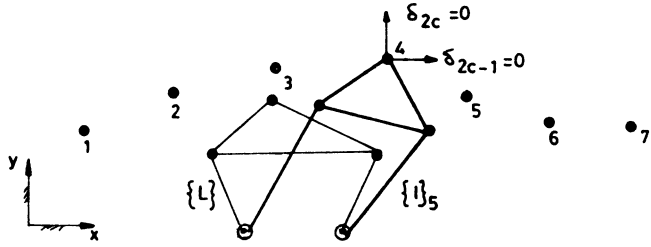


FIGURE 3

The conditions we need to impose in the case presented in figure 3, are the following ones:

$$\delta_{2C-1}=0$$
$$\delta_{2C}=0 \qquad (13)$$

C=number of the node that is made of satisfy the conditions C_d

δ =displacement of the node in the deformated position problem(see ref.[4]).

As we can deduce from what we have exposed, with this method we do not need to know the value of the input parameter in every position; and that is due to the fact that the objective function $\Phi(\{L\})$ is not based on any kind of geometrical error.

If we want to achieve a synthesis with constraint in the input link the mechanism would have to be discretized according to the constraint. We obtain this just as it is shown in figure 4, where the node A, is made to displace itself in the direction of θ_j parameter (the one corresponding to j position) defines. In this case, the constraint condition corresponding to the synthesis that we must impose to the solution of the deformated position problem are as follows:

$$\begin{aligned}\delta_{2C-1} &= 0\\ \delta_{2C} &= 0\\ \delta_{2A} &= 0\end{aligned} \qquad (14)$$

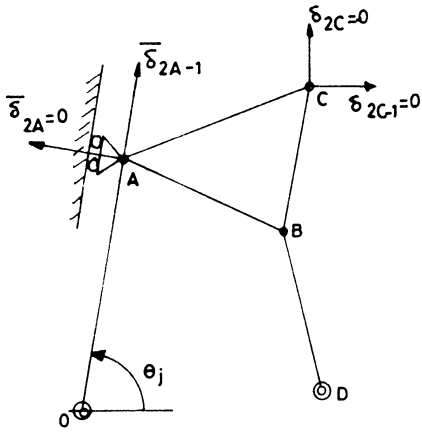


FIGURE 4

As we can see in fig. 4, the third condition is refered to a local $(\bar{x},\bar{y})$ system in the node A that is defined in function of the input parameter θ_j.

Refering to the simultaneous generation of several trajectories, we only have to discretize the mechanism following the suitable way. We also would have to solve the deformated position problem (ref. [4]) according to the constraint conditions corresponding to the synthesis.

FUNCTION GENERATION SYNTHESIS

This synthesis consists in the co-ordination of the positions of the input and output links of a mechanism.

This problem is quite different from the coupler-curve synthesis problem, but, nevertheless, the problem is solved in a very similar way to the former one, using the focusing of the method we present here. In this case, the data we would have to know will be about the S synthesis positions, the value of the initial parameter θ_j and the value of the last parameter β_j. The mechanism would have to be discretized following the way that is represented un figure 5.

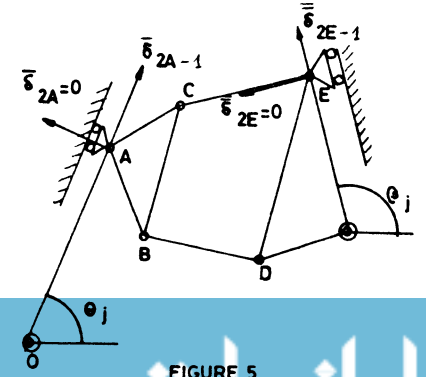


FIGURE 5

The constraint conditions to solve the deformate position problem are:

$$\begin{aligned}\bar{\delta}_{2A}&=0\\ \bar{\delta}_{2E}&=0\end{aligned} \qquad (15)$$

Mixed synthesis (coupler-curve synthesis and function generation synthesis in a same mechanism) could be done easy, only

discretizing the mechanism in the proper way.

RIGID-BODY GUIDANCE SYNTHESIS

This synthesis studies the problem of placing a coupler element of a mechanism in a series of defined positions.

This synthesis shows many similarities with the coupler-curve one, amd only differs from this last one in the fact that now it is necessary to specify S positions, each one of them involves the position of two (or three) nodes among which a constant distance constraint exists. This synthesis may be combined, obviously , with anyone of the formers, taking into account that the discretization must be done according the synthesis being achieved . Figure 6 shows a mechanism that has been discretized for the rigid-body synthesis.

Usually, it is interesting to know the value of parameter θ_j for each one of the positions j , even though it is obvious that it is not necessary from the point of view of the proposed method.

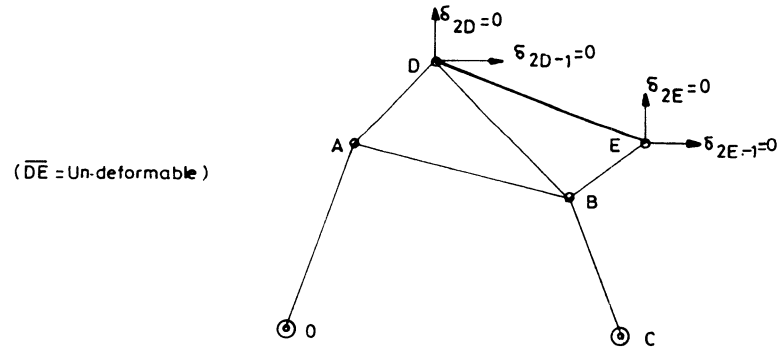


FIGURE 6

In the next pages, we present a Diagram of the Synthesis program, and also of de complete program MECPLAN which allows the synthesis and analysis of plane mechanisms with lower pairs.

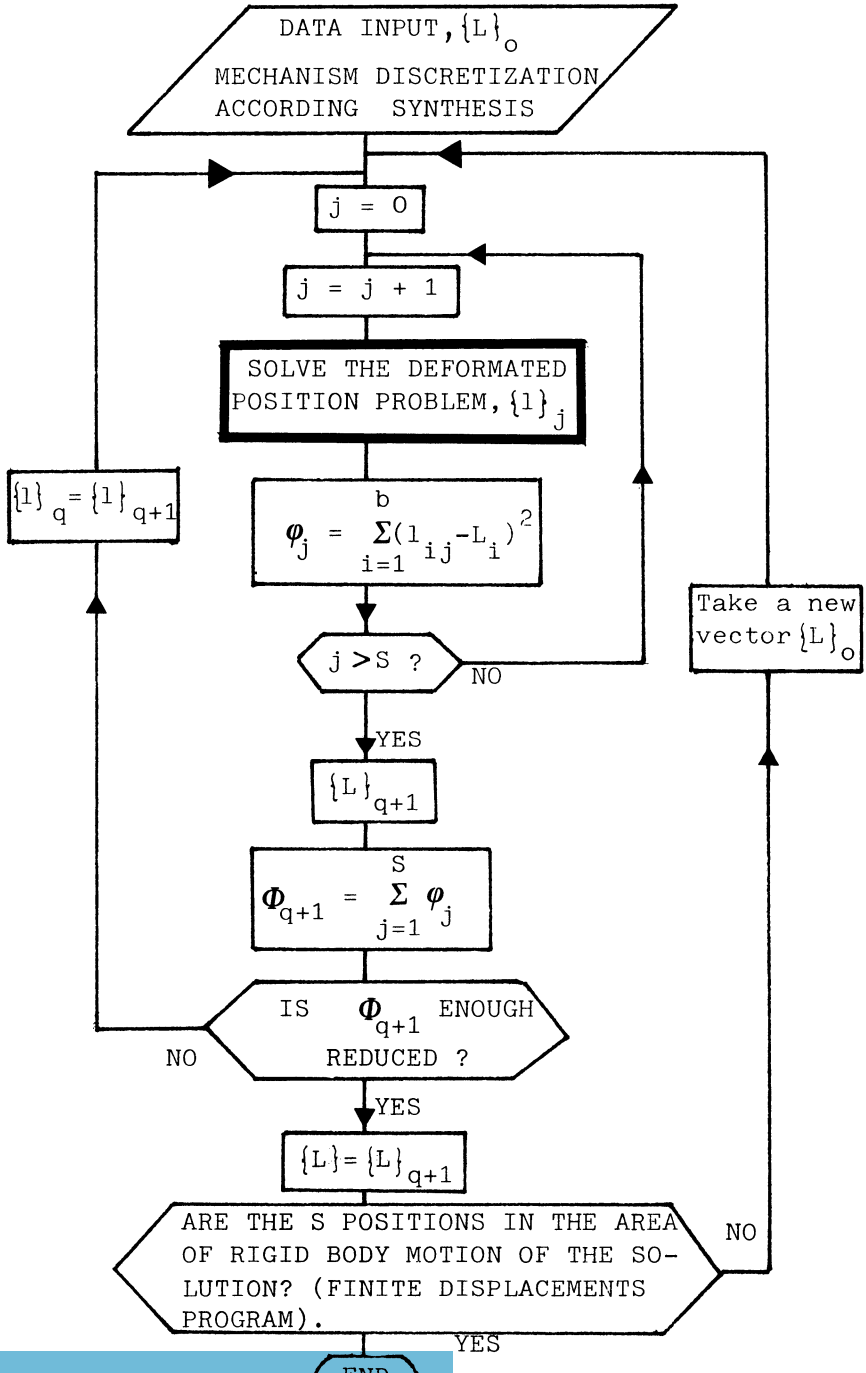


Figure 7.

CONCLUSIONS

In this work we have presented a new method for the optimum synthesis of plane mechanisms. This method uses the same objective function for every kind of synthesis we have presented, even mixed synthesis. Consequently, the computer programs are extraordinarily easy (see figure 7). On the other hand, the synthesis program can be integrated perfectly into the joint of programs of solution of non-linear problems in plane mechanisms, since the nucleous of the synthesis method consists just in the solution of the deformated position problem.

Due to the simplicity of the proposed method, MECPLAN programs for the mechanisms design (ref.[4]) have an homogeneity so that it is possible its implementation even in a desk-minicomputer (MECPLAN-B program in BASIC). Another important remark about the method is that it does not requieres that the initial mechanism generates a charasteristic close to the one we want, just as we can check in the examples; this is due to the quick descent of Φ during the first iterations in the minimizing process. We have to say, nevertheless, that this method has limitations just as those methods using optimitation techniques; among them, we have to remark the fact that the obtained solution depends directly on the initial dimensions $\{L\}$.

In figure 7 we can watch how we need to check that the S synthesis positions can be reached with rigid body motions of the mechanism with the solution dimensions $\{L\}$, before being able to affirm that this mechanism is valid; if this is not the case, it will be necessary to take again a new position, and repeat again all the process.

It is also necessary to remark that the optimun-synthesis method we have described is completely compatible with the kinematic analysis method that is described in references [4] and [5], since it is used as basic concept the geometrical matrix [G] (see figure 8)

Finally, we have to add that, according to what we have already said, it is not necessary to know as a datum the position of all the nodes that are joined to the fixed link; quite the contrary it may be done

a discretization of the mechanism so that it lets the suitable motions to make the synthesis with the minimun number of restrictions (see the examples).

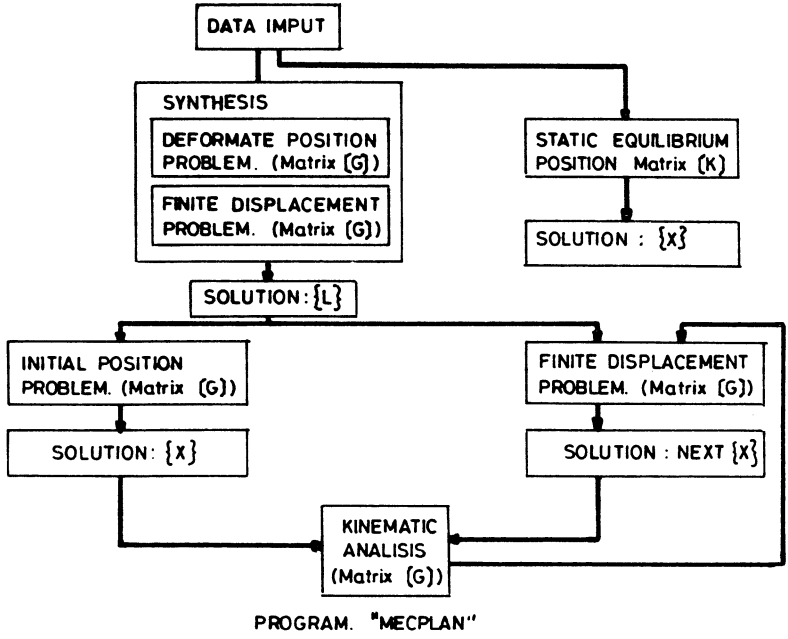
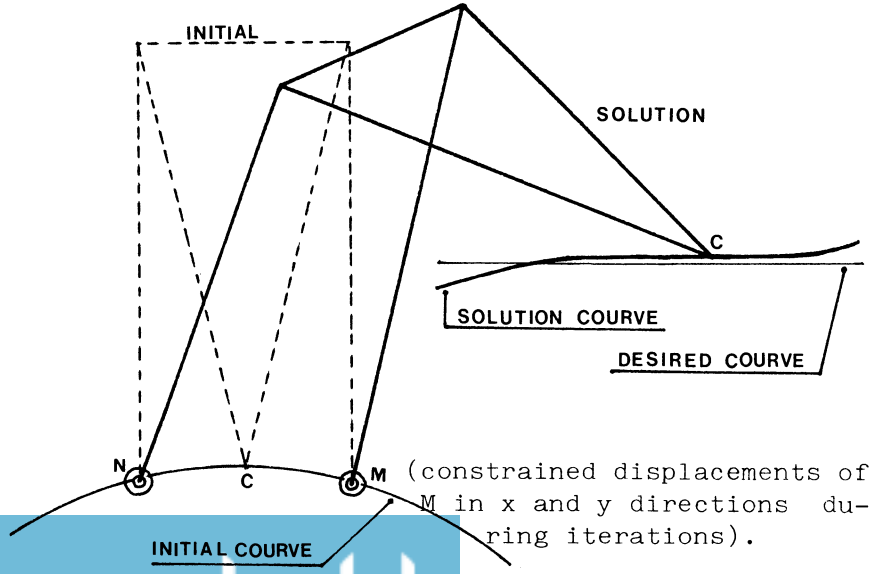
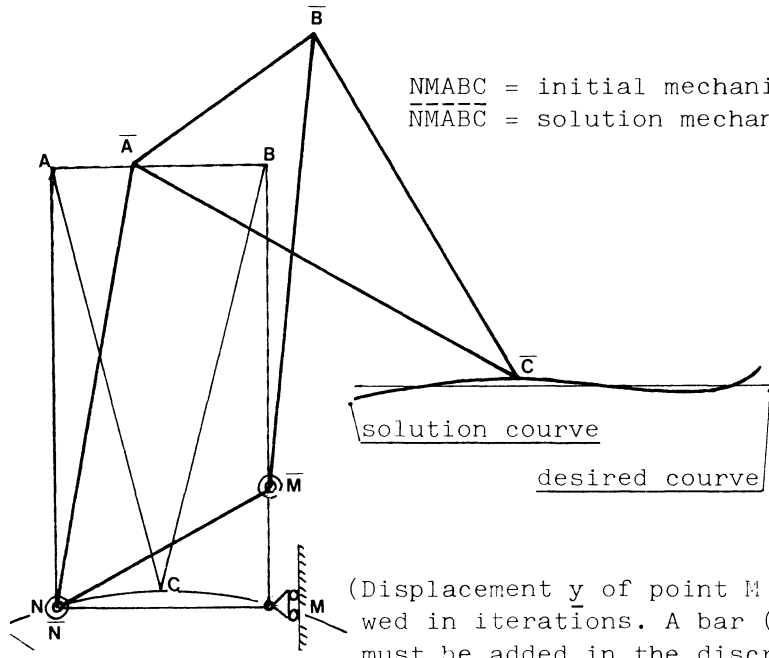


FIGURE 8

EXAMPLES



(constrained displacements of M in x and y directions during iterations).



(Displacement y of point M allowed in iterations. A bar (H) must be added in the discretization)

REFERENCES

1.- Han,C.J.(1966) A General Method for the Optimun Design of Mechanisms. J.Mech.,1,pp.303-313.

2.- Rao,A.C.(1980) Optimum Sy thesis of a Slider-Crank Mechanism using Geometric Programming. Int.J.for Numerical Methods in Engineering, Vol.15,pp.1595-1602

3.- Klein,B.(1981) Zum Einsatz nichtlinearer Optimierungsverfahren zur Rechnerumterstützen Konstruction Ebener Koppelgetriebe.Mechanism and Machine Theory,Vol.16,nº5,pp.567-576

4.- Avilés,R.;Ajuria,Mª.B.;Tárrago,J.A.(1983).A General Program for the Solution of Non-linear Problems in Kinematic Analysis of Mechanisms. (also in this Congress).

5.- García de Jalón,J.;Avilés,R.;Serna,M.A.(1981) Kinematic Analysis of Mechanisms with Matrix Structural Analysis Software.Proc.II Congress ENGSOFT.London.

A GENERAL PROGRAM FOR THE SOLUTION OF NON-LINEAR PROBLEMS IN KINEMATIC ANALYSIS OF MECHANISMS.

R.Avilés; Mª.B. Ajuria; Tárrago,J.A.

Escuela Superior de Ingenieros Industriales.Bilbao Spain

ABSTRACT

In this paper we present a new method for the solution of some non-linear problems in plane lower-pair mechanisms.This method is based in considering the mechanism built up with binary links (bars) with revolute (R) pairs, higher order links can be formed connecting bars in such a way that no relative motion is allowed between them; prismatic (P) pairs are also considered.Four different position problems are solved in this work: initial position, finite displacement,static equilibrium and deformated position.The theorical background,software implementation and examples are presented.

INTRODUCTION.-

In contrast to the determination of velocities and accelerations, the position problems are highly non linear, being this the cause of the greatest difficulties of this problems.The importance of the position problems lies on its solution being the base for kinematic and dinamic analysis, and even an essential point in the synthesis.

Under the name of "position problem" several different problems are included which tradicionally have been reduced to three: initial position, finite displacement and static equilibrium position. In this work, we propose another one, which we could denomina

te as "deformated position problem", and which main interest lies on being the base of an optimum synthesis method.

The initial position problem deals with the determination of the positions of the links of a mechanism when knowing the lenghts of all the bars composing the mechanism,and the position of the input links (as many as degrees of freedom).Finite displacement problem is closely related to the former and it consists in obtaining the consecutive positions of the elements of a mechanism when the input links keep moving according to a certain sequence.Static equilibrium position problem deals with the determination of the equilibrium position of a mechanism made of rigid bars, but including some resorts, under the influence of a certain system of exterior actions. Refering to the deformated position problem, it consist in obtaining the position of the elements of a mechanism when one or more of its points and/or elements are made to fulfil some geometric conditions placed out of the field of rigid-body displacements of the mechanism.

There are different methods to obtain the solution to the position problem that are based on different ideas and that use different iteratives procedures, being the most common the Newton-Raphson and Davidon -Fletcher and Powell (D.F.P) algorithms [1].Brat& Lederer [2] and Paul &Krajcinovic [3] use the Newton -Raphson method and Hall,Root & Sandgren [4] use the D.F.P.algorithm. We can find in reference [5] a more complete version of the solving methods of the position problems.

The method we are going to use in this work is a quasi-Newton method described by Strang (see reference [6]). The method we present has a guaranteed convergence to the solution not depending on the starting position. In order to check the convergence, it has been implemented a subroutine for the random generation of node coordinates. In this way, the starting position is defined without any subjectiveness. Many examples have been analized with this method, which

implementation is very simple .In this sense we should point out that nodes coordinates have been randomly generated in areas of 60×80 cm. while the solution mechanism is cointened in an area of 10×12 cm.,and even in this extreme cases the method has led us to a correct solution.

We are going to describe next the way we can discretize a certain mechanism to analise it using the method we have describe.

DISCRETIZATION OF MECHANISMS

A mechanism is completely defined using two kinds of data: topological and geometric ones. Among the topological data we can include the kinematic configuration in which we find the information about the kinematic pairs, and its relation with the elements. The geometry can be defined by the lenght of the bars composing the mechanism (not solved initial position problem), or by the coordinates of its kinematic pairs (solved position problem).

There are different ways of discretization of mechanisms; in this point we have to remember that from a kinematic point of view, we do not take into account the massic and resistent properties of the elements composing a mechanism. On the other hand it is obvious that the main difference between structure and mechanism is the possibility that this last one has,to move as a rigid solid. Because of that the stiffness matrix of a mechanism is positive semi-defined (a detailed work about this and other differences,and also about the consequence can be found in reference [7]).

Among the many possibilities of discretization,we are using in this method a common one for the matricial analysis of structures, and it consist in decompose the mechanism into bars (binary elements). The basic kind of joint adopted is pair R,and so the kinematic constraints corresponding to pair P will be put in the resulting system as constraint condition. On the other hand , the coordinates of the N nodes composing the mechanism will be taken together in a vector $\{x\}$ of 2N dimension.

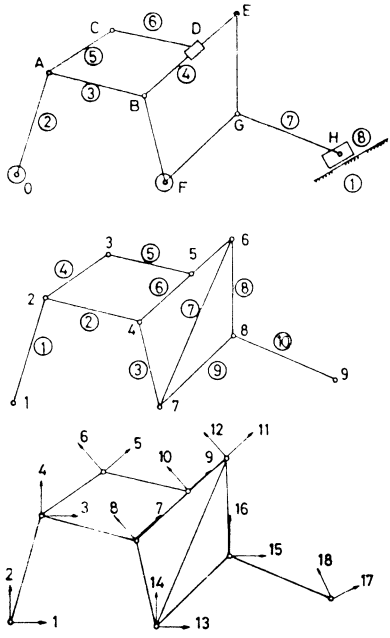


Figure 1.

In figure 1a it is showed a mechanism with one degree of freedom (Grübler) formed by 8 kinematic elements. In figure 1b ,we can see the way it is discretizated using ten bars; and in figure 1c we point out the nodal degrees of freedom,eighteen on the whole,from which five cannot displace and other two, δ_6 and δ_{10}, are related to δ_8 and δ_{12}.

GEOMETRIC MATRIX

According what we have already said,nowadays there are several numeric methods for the analysis of mechanisms(and very few for its synthesis),leting us affirm that none of them uses as a base the properties of the stiffness matrix.

The stiffness matrix of a system contains the information about its capacity to resist loads.But it contains geometric information and even kinematic information (reference[7]).The stiffness matrix of a biarticulated bar i is a symmetrical(4x4) matrix , and it can be designated by k_i .In a system formed by N nodes , the stiffness matrix K is a (2N x 2N) matrix, composed from the sum of the matrices of elements expanded to [K] dimension;we will denominate these expanded matrices with the expresion $[\bar{k}_i]$.Being b the number of bars composing the system , we can write:

$$[K] = \sum_{i=1}^{b} [\bar{K}_i] \qquad (1)$$

Nevertheless, as we have said, in this work we are

not interested (in principle) in the resistent properties of the systems .Expression number (1) can be written in this way:

$$[K]= \sum_{i=1}^{b} \frac{E_i . A_i}{L_i} [\bar{g}_i] \qquad (2)$$

E_i = modulus of elasticity
A_i = section
L_i = lenght of the bar i

Matrix $[\bar{g}_i]$ from equation (2) just depends of the position of both nodes of the bar i, so we will name it "geometric matrix" of the bar i.Beginning from matrix $[\bar{g}_1]$ we can form the geometric matrix of system [G] as:

$$[G]= \sum_{i=1}^{b} [\bar{g}_i] \qquad (3)$$

Both matrices [K] and [G] have the same geometric contents,so it is logic to think that they also have the same kinematic contents.We can verify this establishing that the null-space of both matrices is the same (see reference [7]). Let us have:

$$[K]\{\delta^r\} = \{\emptyset\} \qquad (4)$$

where $\{\delta^r\}$ represents a rigid-body movement of the mechanism,because of this, each bar could move without deformation, so:

$$[\bar{g}_i]\{\delta^r\} = \{\emptyset\} \qquad (5)$$

and adding the b equations type (5):

$$[G]\{\delta^r\} = \{\emptyset\} \qquad (6)$$

so we have proved that both null-spaces are the same.

The advantages of using matrix [G] is that we make much easier to incorporate new data,apart from having the same numeric properties as matrix [K] among which we can mention that one of being band and symmetrical,so the dimension of the matrices

we must work with is reduced to (2N×B) being B the bandwith of the symmmetrical matrix.Every computer program correspondig to this method,MECPLAN- B in BASIC and MECPLAN-F in FORTRAN,have been made using a band matrix $[G_b]$ of dimension (2N x B).

POSITION PROBLEMS.

Using the method we have proposed here,the initial position problem,finite displacement problem,and deformated position problem,can be seen from the same point of view with very few differences among them. The static equilibrium position problem shows some differences respect to other three.First of all we are going to describe the way of solving the first three problems (there are no external action) and,next, the way of solving the static equilibrium problem.

Initial position problem.

This problem may be approached in the following way: let's have the mechanism in a position $\{\tilde{x}\}$,so that we are just sure that the input elements are correctly placed.In this position it is obvious that the lenght of the bars won't be the same ones we've got as data of the problem.Let's have vector $\{L\}$ of the b lengths from the data, $\{l\}$ would be the vector of the lenghts obtained from the starting position$\{\tilde{x}\}$; the problem is just to find a solution position $\{x\}$ (we can find several ones)so that the lengths of the bars are the correct ones.

The problem we have described is strongly non linear and its solution will require an iterative method,so that we must minimize a certain function:

$$\varphi = \sum_{i=1}^{b} (l_i - L_i)^2 \qquad (7)$$

Obviously the minimum value of φ would be zero in the initial position problem and to every step of the finite displacement and bigger than zero in the deformated position problem.

It's easy to demonstrate that:

so that equation (7)can be written in the form:

$$\varphi(\{x\}) = \sum_{i=1}^{b} \{x\}^T [\bar{g}_i] \{x\} - 2 \sum_{i=1}^{b} L_i (\{x\}^T [\bar{g}_i] \{x\}) + \sum_{i=1}^{b} L_i^2 \quad (9)$$

We are going to use a quasi-Newton minimization method (ref.[6]) ,it consist in solving:

$$[J]_q (\{x\}_{q+1} - \{x\}_q) = - \{\nabla\varphi\}_q \quad (10)$$

Being $[J]$the matrix od second derivatives of φ, and $\{x\}_{q+1}$ the q+1 aproximation to the munimum.

We are going to verify the convenience of putting the gradient vector in function of a certain vector $\{A\}$ equivalent to half the former vector.

$$\{\nabla\varphi\} = 2 \{A\} \quad (11)$$

being

$$\{\nabla\varphi\} = 2 \sum_{i=1}^{b} (1 - \frac{L_i}{l_i}) [\bar{g}_i] \{x\} \quad (12)$$

Indeed, in order to obtain the value of $\{\nabla\varphi\}$ (or of $\{A\}$)it is not necessary to do all the operations in equation (12).Let's have a certain bar $\underline{i}$,that in a step of the iterative process,has a lenght l_i.We can see this problem like a problem of a system formed by elastic bars in equilibrium under the action of a certain system of "external actions".We can suppose that ,if bar $\underline{i}$ which initial lenght was L_i has now a lenght of l_i that's because the existence of some actions in the nodes that produce the deformation, (figure 2)

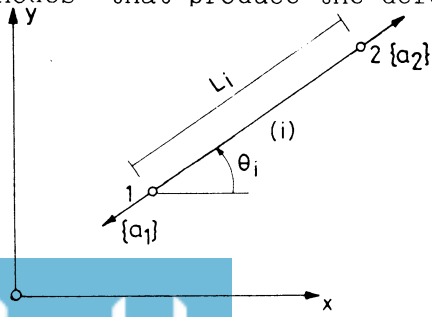


Figure 2.

From figure 2 we can deduce:

$$\{a_i\} = \begin{Bmatrix} \{a_i^1\} \\ \{a_i^2\} \end{Bmatrix} = (L_i - l_i) \begin{Bmatrix} -\cos_i \\ -\sin_i \\ \cos_i \\ \sin_i \end{Bmatrix} = (L_i - l_i)\{h_i\} \qquad (13)$$

we can easily demonstrate that:

$$[\bar{g}_i]\{x\} = l_i\{\bar{h}_i\} \qquad (14)$$

so we have that:

$$\{\bar{a}_i\} = (1 - \frac{L_i}{l_i})[\bar{g}_i]\{x\} \qquad (15)$$

and joining the vectors $\{\bar{a}_i\}$ we obtain a vector, that we can name "actions vector" $\{A\}$

$$\{A\} = \sum_{i=1}^{b} \{\bar{a}_i\} = \sum_{i=1}^{b} (1 - \frac{L_i}{l_i})[\bar{g}_i]\{x\} \qquad (16)$$

Equations (11) and (16) are very interesting because they indicate that to obtain the gradient is not necessary to do the operations indicated in equation (12). Instead of that is enough to calculate the vectors $\{\bar{a}_i\}$ of each bar and joint them to form vector $\{A\}$; this save us a considerable amount of computational effort, even more if we take into account that vector $\{\nabla\varphi\}$ must be obtained in each iteration.

The same way, we can verify that:

$$[J] = 2\sum_{i=1}^{b} (1 - \frac{L_i}{l_i})[\bar{g}_i] + 2\sum_{i=1}^{b} \frac{L_i}{l_i^3}[\bar{g}_i]\{x\}\{x\}^T[\bar{g}_i] = 2[G] \qquad (17)$$

and so, equation (10) can be written in this way:

$$[G]_q(\{x\}_{q+1} - \{x\}_q) = -\{A\} \qquad (18)$$

in order to find the new position $\{x\}_{q+1}$ nearer the minimum than $\{x\}_q$ the following equation must be satisfied:

$$\varphi(\{x\}_{q+1}) < \varphi(\{x\}_q) \qquad (19)$$

When condition (19) is not verified we change the new position ; we can have as a new position the result of correcting $\{x\}_{q+1}$ obtained from (18):

$$\{\tilde{x}\}_{q+1} = \tfrac{1}{2}(\{x\}_q + \{x\}_{q+1}) \qquad (20)$$

and so on until we have $\varphi(\{x\}_{q+1}) < \varphi(\{x\}_q)$. This correction has shown itself very useful, mainly when working with mechanism of a high number of bars and the starting position far away from the solution.

Everything we have said up to this point is valid, not only for the initial position problem, but also for the deformated position .Nevertheless, when we have an initial position problem (and in every positon in the finite displacement problem) the iterati ends when we reach a $\{x\}$ position where : $\varphi(\{x\}) < \epsilon$, being ϵ a quantity small enough (in MECPLAN programmes it has been taken $\epsilon = 10^{-7}$)on the contrary, in the deformate position problem we don't know the value of φ_{min} , so at first we wouldn't be able to know if the minimum has been reached in a certain moment.

Deformated position problem.

The knowledge of the moment in which the minimum of $\varphi(\{x\})$ is reached can be obtained from the intuitive interpretation of the problem. Indeed, in the solution position the actions on the non-constrained nodes are null. Calculating in each iteration:

$$\xi_{q+1} = \{A\}_q^T(\{x\}_{q+1} - \{x\}_q) \qquad (21)$$

The solution position will be reached when $\xi_{q+1} < \epsilon$

Figure 3 illustrates a typical deformated position problem. Point C of the coupler is made to have a position C_d placed out of the field of rigid body displacements of the mechanism (as we have already said, the interest of establisihing this problem

lies in its being the base of an optimum synt method, that is described in reference [8].

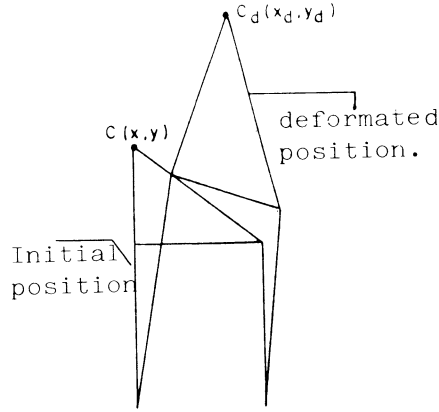


Figure 3

This is, to solve the deformated position problem, it's enough to considerate the geometric restriction we want to satisfy as a constraint condition in the system of expression (18).

It's obvious that condition (21) is also verified when the minimum value of φ is zero, this is, when C_d belongs to the field of rigid body movements of node C of the mechanism (initial position problem).

Finite displacement problem.

As we have already said, it consist in obteining the cosecutive positions of the elements of a mecha_ nism when the input elements move according to a certain sequence. From the definition we can see its close relation to the initial position, so we tend to solve it with the same procedure we have already explained, taking as an starting position for each step the solution of the previous step. (See the examples).

The diagram in Figure 4 represents in a schematic way the solution process for both, the initial posi_ tion problem and the finite displacement problem.

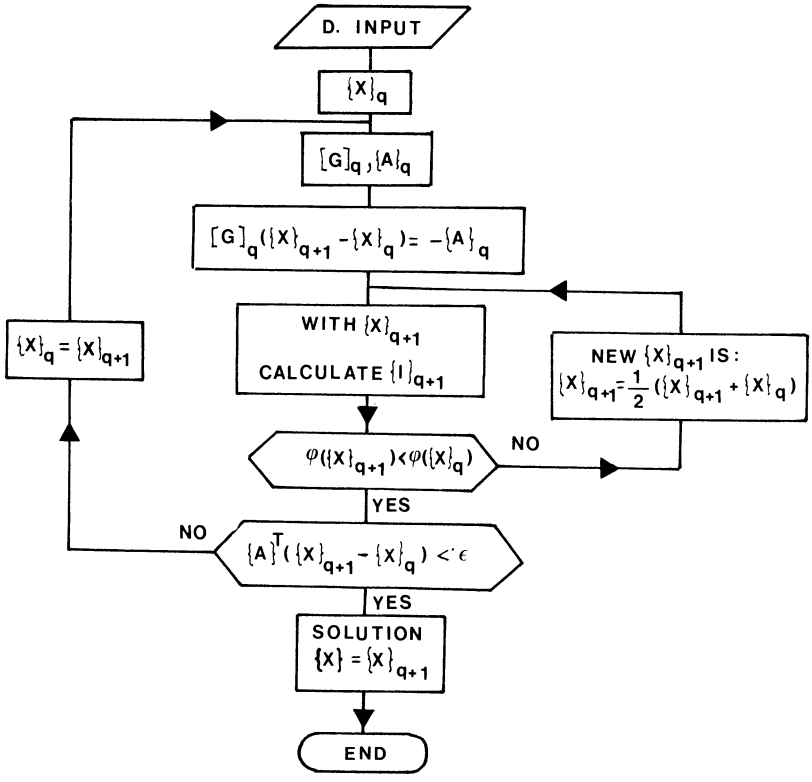


Figure 4

Static equilibrium problem.

It s defined as the one used to obtain the equilibrium position of a mechanism including elastic elements when it is subjected to certain exterior actions $\{F_e\}$

The static equilibrium should correspond to the minimum of a certain function η that consist of two terms :one corresponding to the exterior actions and the other corresponding to the elastic deformation energy of the resorts or elastic elements:

$$\eta = \tfrac{1}{2} \sum^{r} k_{ab}(l_{ab} - L_{ab})^2 - \{x\}^T\{F_e\} \qquad (22)$$

k_{ab}= stiffness of the resorts that joins nodes ab
l_{ab} = distance between a and b
L_{ab}= initial length of the resort ab (linear and

extension resort)
r = number of resorts
x = the position vector we are trying to calculate

The solution will be position $\{x\}$ so that has its minimum value, and at the same time $\varphi = 0$ refering to the rigid part of the mechanism. From a mathematical point of view it is a problem that can be solved by minimizing:

$$\Psi(\{x\}) = \tfrac{1}{2}\sum^{r} k_{ab}(l_{ab}-L_{ab})^2 - \{x\}^T\{F_e\} + \lambda\sum_{i=1}^{b}(l_i-L_i)^2$$

being λ a great number. (23)

The same way it happened in the minimization of $\varphi(\{x\})$ using the same quasi-Newton method to obtain the minimum value of η we otain:

$$[J_r]_q(\{x\}_{q+1} - \{x\}_q) = -\{\nabla\eta\}_q \qquad (24)$$

being

$$\{\nabla\eta\} = \sum^{r}(1-\frac{L_{ab}}{l_{ab}})[\bar{k}_{ab}]\{x\} - \{F_e\} \qquad (25)$$

As we have already seen, the first term of equation (25) is the vector of the actions between the nodes a and b:

$$\{\nabla\eta\} = \sum^{r}\{\bar{f}_{ab}\} - \{F_e\} = \{F_r\} - \{F_e\} \qquad (26)$$

The same way we also obtain

$$[J_r]_q = \sum^{r} k_{ab}[\bar{g}_{ab}] = \sum^{r}[\bar{k}_{ab}] = [K_r] \qquad (27)$$

replacing in the equation (24) we obtain:

$$[K_r]_q(\{x\}_{q+1} - \{x\}_q) = -\{F_r\}_q + \{F_e\} \qquad (28)$$

According to equation (23) and equation (18) (minimum of $\varphi(\{x\})$), the minimum value $\Psi(\{x\})$ would be obtained from:

$$([K_r]_q + [G]_q)(\{x\}_{q+1} - \{x\}_q) = -(\lambda\{A\}_q + \{F_r\}_q) + \{F_e\} \qquad (29)$$

Equation (29) can be expressed in a more compact way:

$$[K]_q(\{x\}_{q+1}-\{x\}_q) = -\{F\}_q + \{F_e\} \qquad (30)$$

where [K] is the stiffness matrix of a system with r elements of stiffness k_{ab} and b elements of stiffness λ, {F} is the internal forces vector. Obviously, the value of λ (the same for all the rigid elements) must be much greater than every value of k_{ab}.

The solution of the static equilibrium problem is this way easily obtained because it only requires to form the stiffness matrix of a system and to obtain vector {F}. The rest of the iterative process is similar to the one we use for the other position problems, the equilibrium position is reached when a certain function ψ_{q+1} becomes minor than a certain ϵ, being :

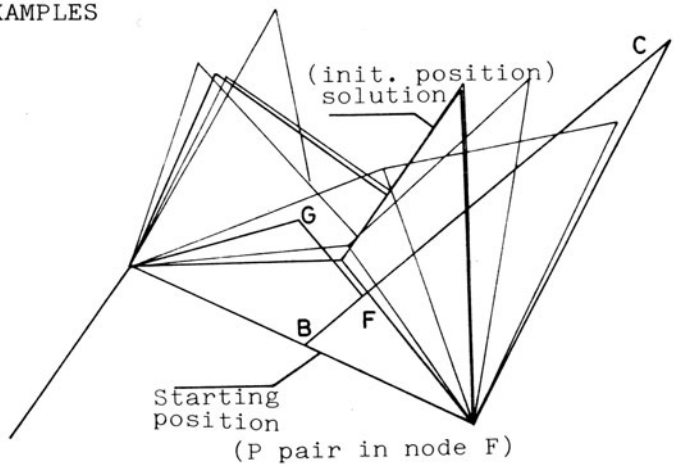
$$\psi_{q+1} = (\{F\}_q-\{F_e\})^T(\{x\}_{q+1}-\{x\}_q) \qquad (31)$$

This way of stablishing the problem makes little differences in the solving method of the different position problems we have already mentioned, allowing us to make really simple and general computer programs. The diagram in figure 4 is valid, with very few modifications for the static equilibrium problem.

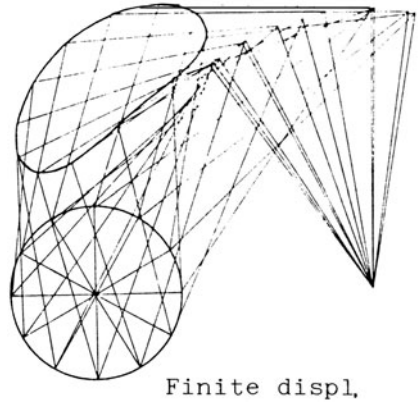
CONCLUSIONS.

In this work, we have shown the solution to 4 important non linear problems within the kinematic analysis of mechanisms. This method is based on the geometrical matrix of a system formed by binary elements with pairs R, the prismatic pairs (P) are taken into account introducing the kinematic restrictions, that set them, apart in the resulting equation system. The most important characteristics of this method, may be, that we don t need a good initial aproximation; the low number of iterations to obtain a solution because of the easy and compact mathematic formulation; it s made very simple the implementation of the method in a digital computer not being necessary the topologic pre-processing labour, that are essential using some methods.

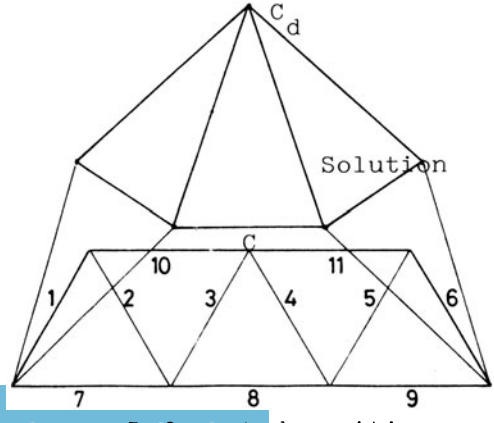
EXAMPLES



(mechanism with 1 d.o.f.)



Finite displ.



Deformated position
(isostatic structure)

REFERENCES

1. Davidon, W.C.(1954) Variable Metric Method for Minimization. Argonne National Laboratory, ANC-5990 Rev. University of Chicago. Chicago, Ill.

2. Brat, V.; Lederer, P.(1973) KIDYAN: Computer-Aided Kinematic and Dynamic Analysis of Planar Mechanisms. Mechanism and Machine Theory, pp. 457 - 467.

3. Paul, B.; Krajcinovic, D.(1970) Computer Analysis of Machines with Planar Motion. Part 1-Kinematics. Journal of Applied Mechanics, pp. 697-702.

4. Hall, A.S.; Root, R.R.; Sandgren, E.(1977) A Dependable Method for Solving Matrix Loop Equations for the Three-Dimensional Mechanism. Journal of Engineering for Industry, pp. 547-550.

5. García de Jalón, J.; Serna, M.A.; Avilés,R.(1981) Computer Method for Kinematic Analysis of Lower-Pair Mechanisms-II: Position Problems. Mechanism and Machine Theory, Vol. 16, nº 5, pp. 557-566.

6. Strang, G.(1980) The Quasi-Newton Method in Finite Element Calculations. Computational Methods in Nonlinear Mechanics. North Holland Pub. Comp.,pp. 451-456.

7. García de Jalón, J.; Avilés, R.; Serna, M.A.(1980) Kinematic Analysis of Mechanisms with Matrix Structural Analysis Software. Engineering Software II, London.

8. Avilés, R.; Ajuria, Mª.B.; Tárrago, J.A.(1983) A General Program for the Optimum Synthesis of Plane Mechanisms. In this Congress.

GRAFFIC - A COMPUTER PACKAGE FOR THE INTERACTIVE GRAPHICAL REPRESENTATION OF FLUID-FLOW PHENOMENA

Nikos C Markatos and Coulis A Pericleous

Concentration Heat and Momentum Limited
40, High Street, Wimbledon, London SW19 5AU

ABSTRACT

GRAFFIC is a three-dimensional graphics computer package which provides interpretation and display facilities for numerical predictions of phenomena involving fluid-flow and heat/mass transfer. It is designed for use as a pre- and post-processor for solution methods of the finite-domain or finite-difference type. GRAFFIC operates interactively, using a storage-type graphics terminal, with associated hard-copy units if required. Communication with the user is via a series of meaningful prompts, designed to make operation self explanatory.

Facilities provided by GRAFFIC include representation of the flow geometry and solution mesh, contour maps and surfaces of scalar fields, and vector fields represented either by vector maps on selected planes or vector lines or streamlines. For two-phase flow computations, GRAFFIC provides separate representation of the velocities and other properties of each phase.

A range of view options are provided including: perspective from a specified viewpoint; oblique views; two-dimensional side, top and end views; and a stereoscopic effect provided by two perspective views at a small displacement.

The paper provides a description of the structure and organisation of GRAFFIC and illustrates its use by showing applications to results obtained from the PHOENICS program.

1. INTRODUCTION

GRAFFIC is a pre- and post-processor for fluid-flow solution methods of the finite-domain or finite-difference type. While GRAFFIC can be connected to any finite-domain or finite-

difference analysis program, the version described in this paper has been designed specifically for application to data fields produced by the PHOENICS computer program (Rosten et al, 1983).

The use of a finite-domain or finite-difference analysis program comprises essentially three stages:

- preparation of problem-specifying information (pre-processing);
- processing it in accordance with the solution procedure; and
- provision of the solution results (post-processing).

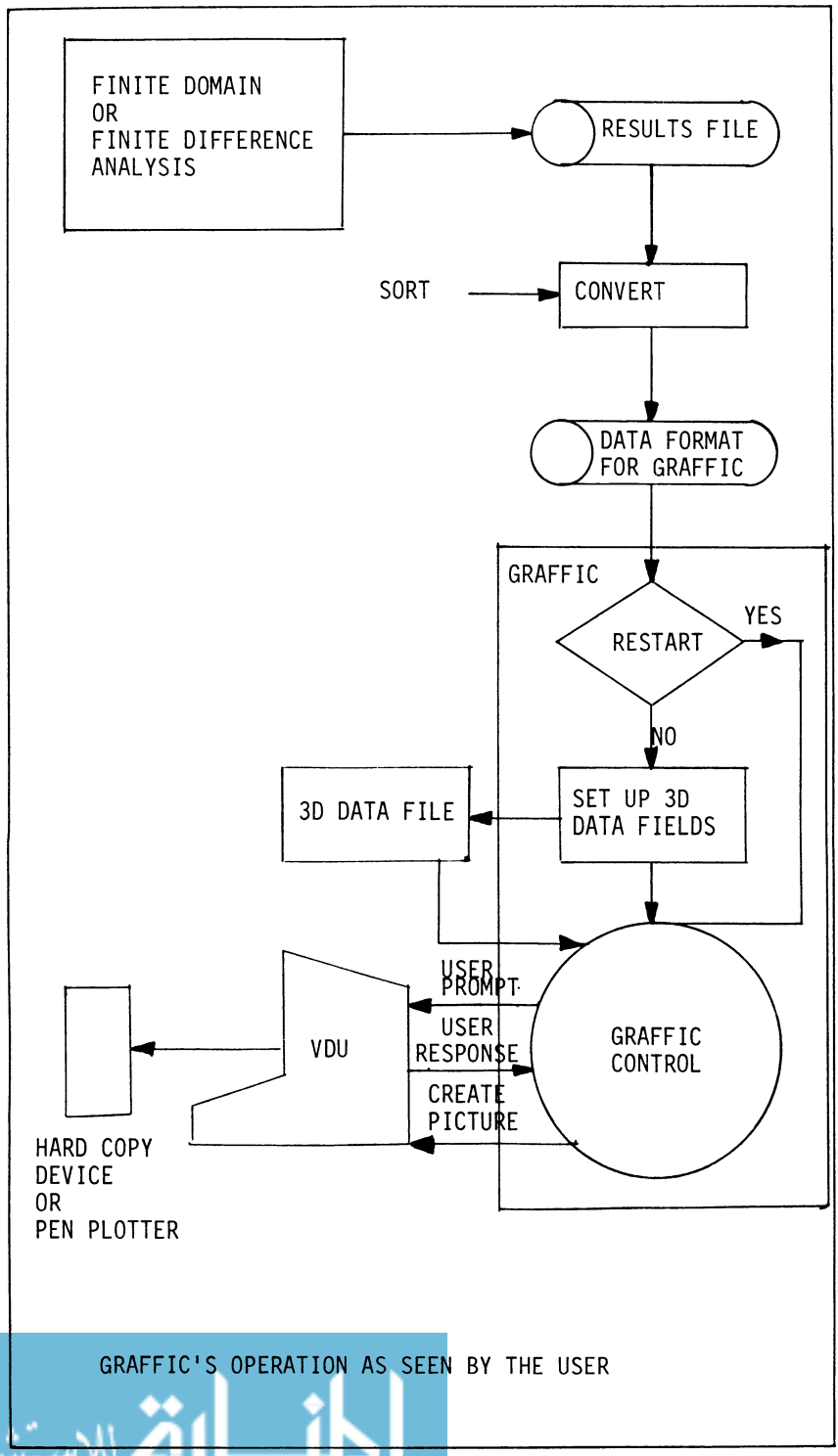
An advanced fluid-flow program such as PHOENICS can be used to predict complex three-dimensional fluid motion and heat/mass transfer, including effects such as turbulence, two-phases and chemical reaction. The amount of data describing these fluid flow fields can be vast and interpretation becomes a significant information processing problem. The use of computer graphics is an obvious approach to the solution of this problem. Indeed, techniques for 2D graphics have long played a significant role in data interpretation. However, direct extension of existing 2D techniques to 3D flow fields is not straightforward since, for example, a scalar having three degrees of freedom is a four-dimensional surface, making the complete portrayal on a single diagram virtually impossible. Furthermore, all display media currently available are two-dimensional, and the many-to-one nature of the required transformations, to create three-dimensional looking diagrams, severely restricts the amount of information that can be presented at any time. Finally, the amount of data describing three-dimensional fluid motion is enough to tax the largest computer system, and necessitates the use of an efficient mass storage data base system. GRAFFIC offers solutions to these problems by combining several methods of 3D data field representation, an efficient data base system and a range of picture generation and editing facilities in a single, interactive computer program.

GRAFFIC can be used both to obtain graphical checks of the input data and for displaying the output solution fields.

2. DESIGN CONCEPTS AND OPERATION

The primary design objective of assisting with the problems of data pre- and post-processing was met by combining the following three elements:

- control of the graphics terminal; this involves the use of TEKTRONIX PLOT10 or other basic graphic software;



GRAFFIC'S OPERATION AS SEEN BY THE USER

- a random-access data base;
- user communications via short, meainingful mnemonics defining the different operations.

The main stages in the use of GRAFFIC for post-processing, as illustrated in Figure 1, are:

- During solution a "Results" file, containing the model input data and the solution results must be created;
- The Results file is then converted into the format required by GRAFFIC via a subroutine called SORT. Because the structure of the Results file differs between finite-domain programs, each will require its own conversion (interface) program. The current version is specifically designed for PHOENICS but others can easily be created;
- GRAFFIC is started by the user from the remote graphics terminal, and a direct-access, 3D solution-field data base is established;
- The GRAFFIC control program then provides a series of prompts inviting the user to enter "key-phrases" which select the next action, resulting in the creation of the picture on the screen. The major options are shown in Figure 2;
- Finally, the user can, if required, create a hard copy of the picture image on the screen using a separate hard-copy device.

3. PROGRAM MODULES

31. Coding Philosophy

A code designed for graphical analysis of 3D numerical predictions of fluid flow phenomena differs significantly from codes furnishing these predictions. The prediction codes consist of complex algebraic procedures, repeated many times over large data fields and therefore make considerable use of the computer central processor. In addition such codes are not interactive and can be submitted as batch jobs with less stringent core restrictions. In contrast, GRAFFIC consists of a multitude of operations, each of which can be regarded as being distinct from all others. In interactive use these operations must be capable of being performed in any order and must therefore involve a minimum of interaction and interference.

In general, the arithmetic requirements for each operation are small, and use in a time-sharing interactive mode imposes core restrictions.

OPTION	TYPICAL USER RESPONSES
• Select Viewpoint (prompt 'VIEW OPTION')	PERSP: Provides perspective view from a specified point SIDE, END, -SIDE, -END: Provide suitable side elevations, and rotate view by 180° OBLIQUE: Provides view from a specified point, without the perspective distortion
• Representation of grid and geometry (prompt 'GEOMETRY OPTION')	CELLS: Draws mesh cells over part or whole of domain MESH PLN: Constructs mesh planes BLOCKAGE, OUTLINE: Represent blocked regions in solution domain
• Select phase to be plotted (prompt 'SELECT PHASE')	PHASE 1 or PHASE 2: Selects first or second phase variables for two-phase results
• Select type of plot (prompt 'SELECT OPTION')	STREAM: Constructs streamlines VECTOR: Constructs vector diagrams on selected planes SCALAR: Represents scalar fields in either of following ways:
The variable to be plotted is selected via user-specified titles stored with the results	MAP: Generates contour maps on selected planes PROFILE: Generates contours on a coodinate surface as 3D profiles SURFACE: Generates the equivalent of a contour map in 3D (a contour surface)
• Manipulate picture (prompt 'PICTURE OPTION')	STORE: Stores pictures as they are generated EDIT: User edits selected picture elements (eg modify texture, eliminate hidden lines, etc)
The user can at any stage return to earlier or jump to later ones	TEXT: Provides specified text on picture COPY: Activates hard copy device MAGNIFY: "Zoom" into picture and examine details REDRAW: Redraws contents of data base onto the screen

FIGURE 2: MAJOR GRAFFIC OPTIONS

As a consequence, GRAFFIC has been written as a family of many subroutines, some of which are very short. Such practice offers many advantages: a) each routine performs a discrete operation and is fully self-contained; b) the use of a subroutine to perform repeated tasks does not increase core requirements; c) it allows use of overlays, further reducing core demands; d) it improves the code's checkability; and e) future extensions can be easily implemented by adding new subroutines.

3.2 The MAIN Program

All primary operations are controlled by a MAIN program, as follows: i) Data initialisation; (ii) geometrical representation; (iii) selection of first and second phase variables; (iv) construction of streamlines; (v) construction of vector maps; (vi) scalar field representation via contours, profiles or scalar surfaces; and (vii) picture generation and management, including the selection of alternative view points.

Each operation, except (iii), is controlled by a primary subroutine, residing in a primary overlay, which calls upon a base of small service subroutines for data management and graphical actions.

3.3 Data Initialisation

Up to 25 variables can be accepted for processing, specified on a Cartesian or cylindrical polar solution mesh. The current structure also allows for extending GRAFFIC to body-fitted, non-orthogonal, coordinates. Once read-in, the variables are stored in a direct-access file, each as a single record. Mass fluxes are then calculated for each control cell face, for both first and second phase velocities. These fluxes are stored, for use in the streamline calculations. GRAFFIC performs an independent mass balance and informs the operator of the location and magnitude of the maximum error in the field. This section is non-interactive and is controlled by the primary subroutine INIT.

Subsequent sessions involving the same data can avoid this initialisation step and restart from the fields stored on the direct-access file.

3.4 Geometrical Representation (Keyword GEOM)

This is controlled by the primary subroutine GEOMTR. It represents various geometrical features of the solution domain, including: (i) the domain outline; (ii) the mesh used; (iii) individual cells or groups of cells; and (iv) the outline of regions in the domain inaccessible to flow (blocked).

3.5 Construction of Streamlines (Keyword STREAM)
This section is controlled by subroutine STREAM. The calculation of streamlines can be performed for steady, incompressible or compressible, single- or two-phase flow fields. Noteworthy features are: (i) a single streamline can be traced; (ii) streamlines are traced in segments, with the option of continuation or termination at each break-point; (iii) streamlines are automatically terminated upon closure, at a stagnation point or if they leave the domain; (iv) families of streamlines can be constructed from a given line; (v) streamlines can be traced backwards; and (vi) inlets and outlets can be drawn to assist in selection of streamlines.

3.6 Construction of Vector Maps (Keyword VECTOR)
This section is controlled by subroutine VECTOR. Features include: (i) velocity vectors can be constructed on any coordinate surface; (ii) default scaling prevents vectors from overlapping; (iii) the default scaling can be overriden by the operator; and (iv) the vector routine generates an absolute-velocity field for contour maps.

3.7 Scalar-field Representation (Keyword SCALAR)
This section is controlled by subroutine CONTUR. Three scalar representations are possible in this section:

(a) Contour maps (Keyword MAP).
These are 2D contour maps on a selected coordinate surface. Noteworthy features are: (i) contour levels can be set automatically over a range specified by the operator; (ii) contour levels can be set manually if required; and (iii) contours can be labelled and current levels can be displayed at the bottom of the screen.

(b) Profiles (Keyword PROFILE)
They are 3D representations of contour maps, selected on any coordinate surface, the third dimension representing the magnitude of the scalar. The scaling of the excursion from the coordinate surface is controlled by the user.

(c) Surfaces (Keyword SURFACE)
A single value of a scalar in a 3D field defines a surface, that can be constructed by GRAFFIC.

3.8 Picture Generation and Management (Keyword PICTURE)

3.8.1. View Selection
The following projections of the solution domain are available: (i) perspective from a selected viewpoint; (ii) oblique, from a selected viewpoint; (iii) side, end and plan elevations; and (iv) stereoscopic, from any viewpoint.

3.8.2 Picture Management

This provides the following features:

(i) Up to six pictures can be displayed together on the screen.
(ii) Individual elements of a picture can be removed or replaced by the user.
(iii) Elements can be smoothed or drawn with dotted lines.
(iv) A zoom-in feature allows enlargement of part of the picture.
(v) Text can be placed anywhere in the picture using a cross-hair cursor.
(vi) A hidden line elimination facility is available for certain types of picture elements.

4. EXAMPLES OF THE USE OF GRAFFIC

Figures 3 to 13 present some typical graphical output obtained by GRAFFIC.

All presented results have been generated by using PHOENICS (Rosten et al, 1983). The figures illustrate the main capabilities currently built into GRAFFIC, including mesh and geometry representation, vector maps, 2D and 3D streamlines, contours, surfaces, 3D profiles, and alternative view options including perspective. Further examples may be found elsewhere (eg Markatos 1981, Markatos and Mukerjee 1981, Malin et al 1982, Markatos and Cox 1982, Seppen 1982, Rosten et al 1983).

Figures 3 and 4 illustrate the GRAFFIC mesh and geometry representation. Figure 3 shows a cylindrical domain with inlets and outlets and selective mesh illustrations. Figure 4 shows a submarine hull with four fins in a cruciform arrangement, represented by porosities.

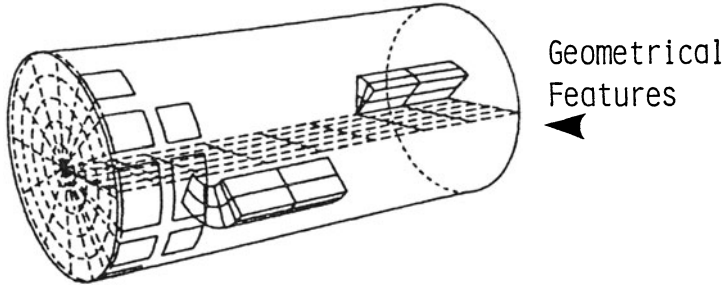


FIGURE 3: GEOMETRICAL FEATURES

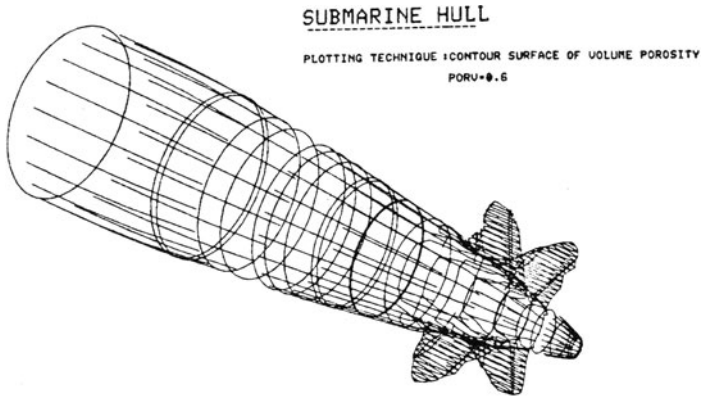
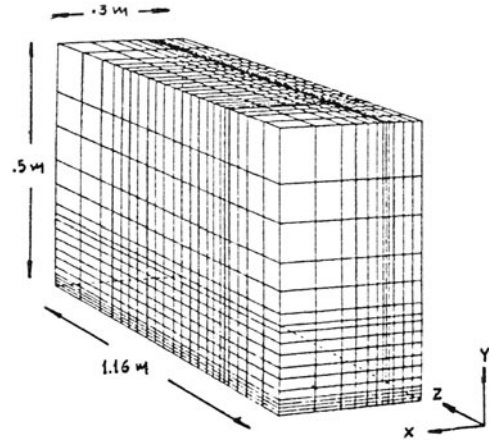


FIGURE 4: GEOMETRY REPRESENTATION OF SUBMARINE HULL



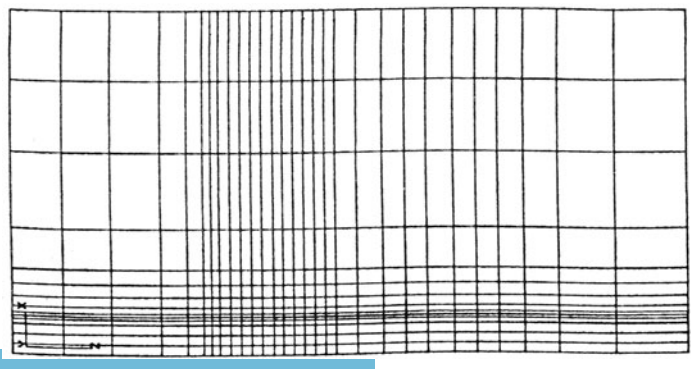


FIGURE 5: THE FINITE-DOMAIN MESH USED FOR CAR SIMULATION

Figures 5 to 8 refer to a study of external motor-vehicle aerodynamics. Figure 5 presents the mesh used in a perspective view and top view.

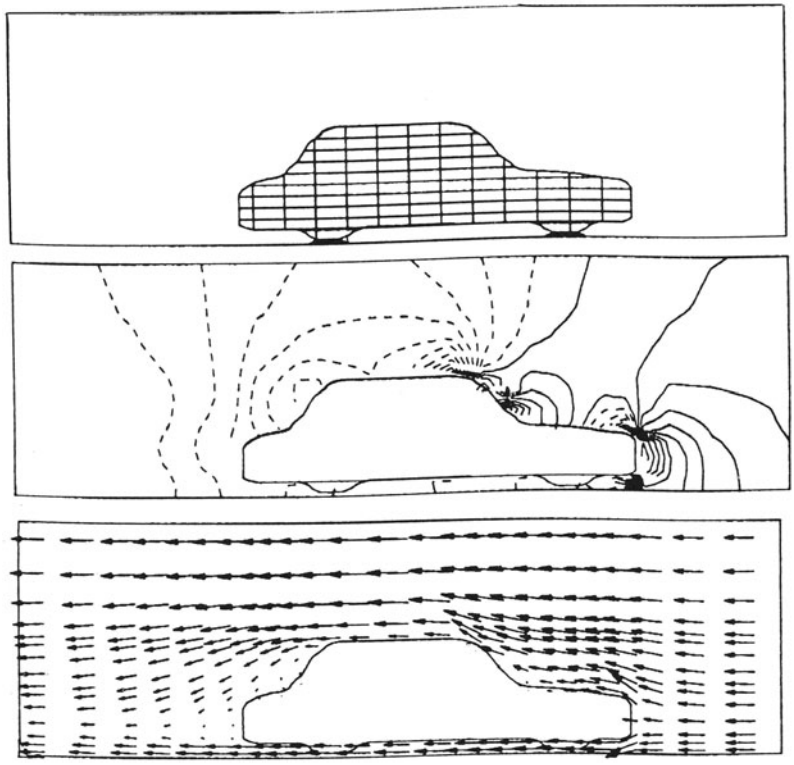


FIGURE 6: CAR SHAPE, PRESSURE COEEFICIENT CONTOURS AND VECTOR MAP AROUND THE VEHICLE (SIDE VIEW)

Figure 6 shows 2D contours of the static pressure coefficient and vector map around the vehicle, and the vehicle shape as represented in PHOENICS by porosities, in a y - z plane near the centre of the car. Positive pressure-coefficient contours are shown as solid lines and negative as dotted lines.

Figure 7 illustrates the capability of GRAFFIC to enlarge any selected region of the domain, and shows the velocity vectors around the rear of the car.

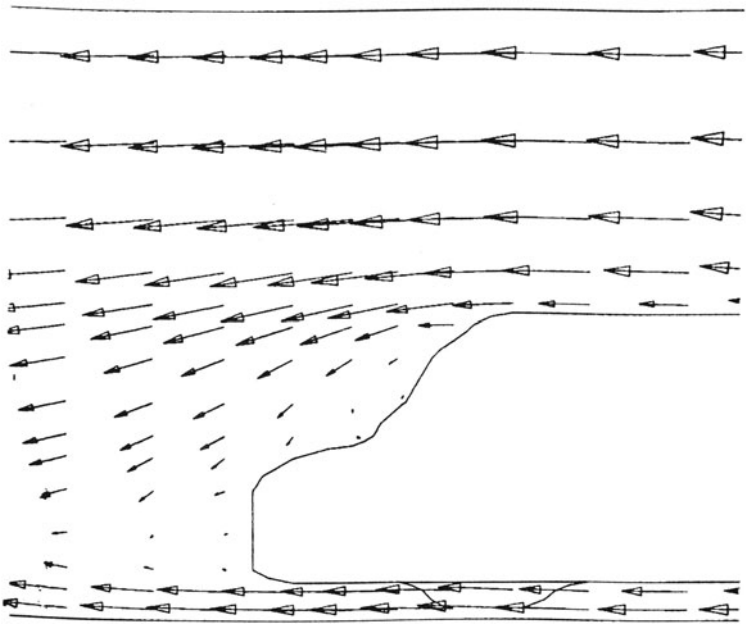


FIGURE 7: ENLARGEMENT OF REGION BEHIND THE VEHICLE

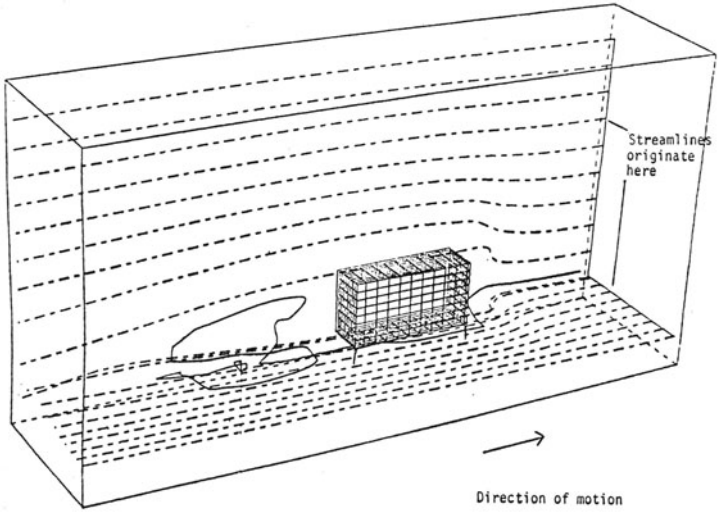


FIGURE 8: PERSPECTIVE VIEW OF STREAMLINES FOR FLOW AROUND A BOX-SHAPED VAN

Figure 8 shows streamlines for the flow around a box-shaped van. The chain-dashed lines correspond to streamlines originating along the vertical and horizontal lines shown. The solid line shows a streamline which becomes trapped in the recirculation zone.

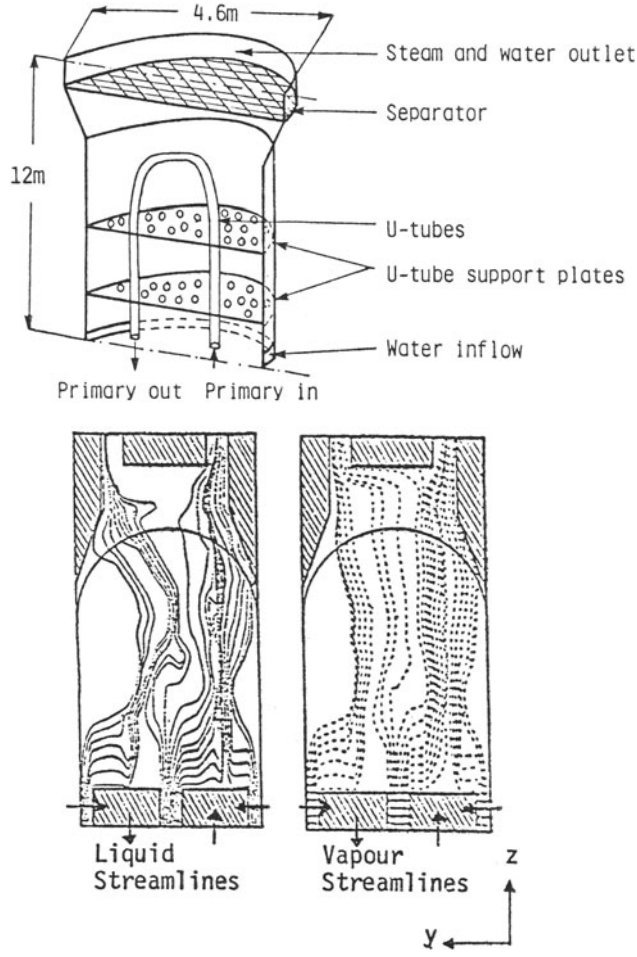


FIGURE 9: THE STEAM-GENERATOR CONSIDERED AND TWO-PHASE STREAMLINES.

Figures 9 and 10 refer to the two-phase application of PHOENICS and GRAFFIC to steam-generators of the kind used in pressurised-water reactors. Figure 9 shows liquid and vapour streamlines, together with the geometry of the generator considered, and Figure 10 surfaces of high steam concentration. The differences between the phase streamlines, due to interphase slip, is apparent and there is clearly an asymmetry in steam concentration between the hot and cold

sides of the generator, as expected.

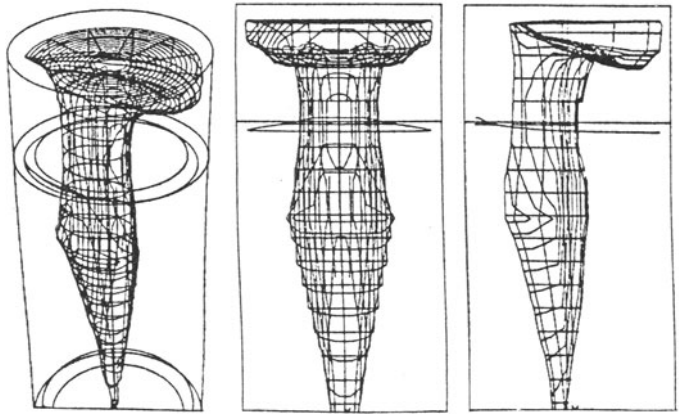


FIGURE 10: SURFACE OF HIGH STEAM CONCENTRATION

Figures 11 and 12 refer to the flow of smoke in enclosures. The particular problem simulated concerns the initiation of fire in one of the shops along the enclosed L-shaped shopping mall, as shown in Figure 11.

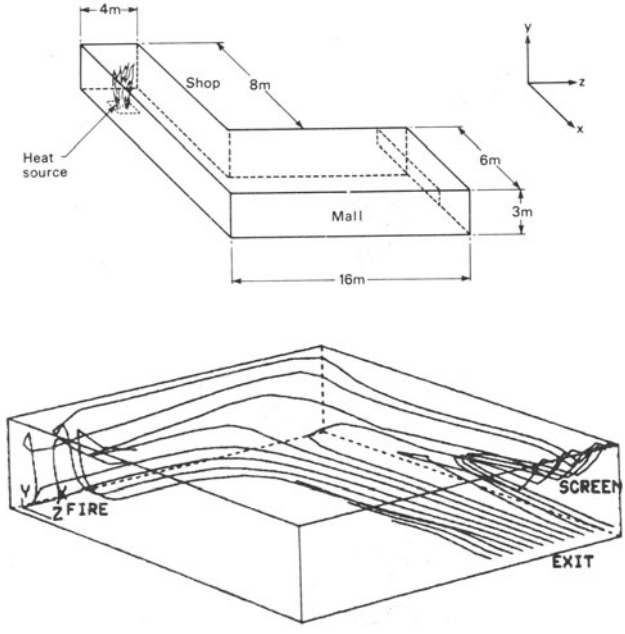


FIGURE 11: THE GEOMETRY CONSIDERED AND PLOTS OF STREAMLINES

Figure 11 presents, in perspective view, plots of streamlines. These plots show the paths of fluid particles originating at particular locations. Ten equally spaced paths are shown originating at the exit near to the floor. It is clear that cold air is drawn into the mall close to the floor, becomes hot in the region of the fire and exits near the ceiling. Figure 12 presents the temperature time history in the form of the T = 600°K surfaces. The spread of smoke between t = 80 and 155 secs is obvious.

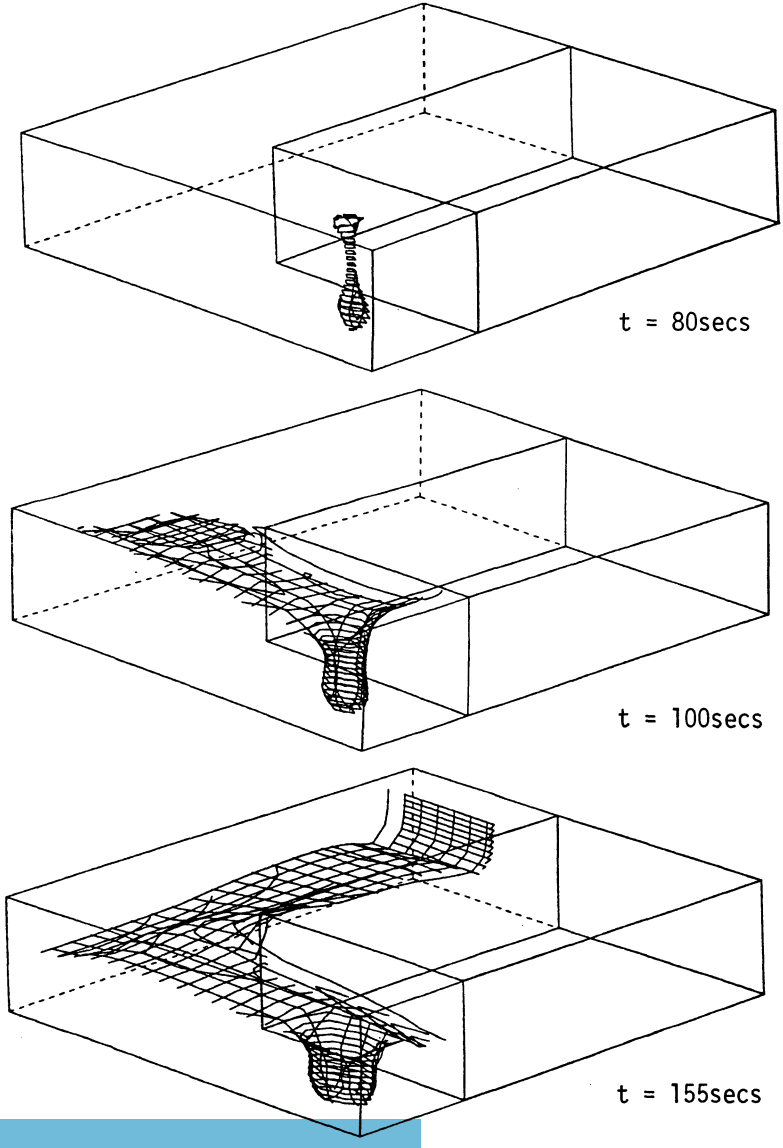


FIGURE 12: TEMPERATURE TIME HISTORY (T = 600°K).

Finally, Figure 13 presents surfaces of water-droplet mass-fraction (5 x 10^{-6}Kg/per Kg of mixture) in the plume exiting to the atmosphere from a cooling tower.

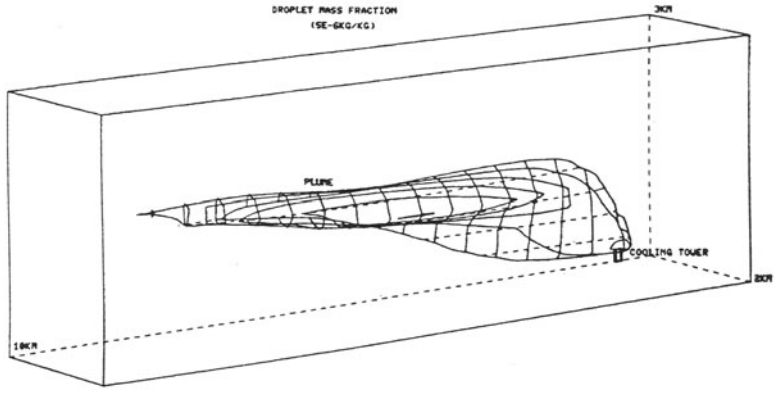


FIGURE 13: SURFACES OF DROPLET MASS-FRACTION IN THE COOLING TOWER PLUME

5. CLOSING REMARKS

A three-dimensional, interactive graphics computer package, GRAFFIC, has been described. It is currently in use at CHAM Ltd., using a Perkin Elmer 3220 mini-computer and a Tektronix graphics terminal, and on a number of other systems, including IBM, Univac, CDC and other machines.

Typical computer-system requirements for GRAFFIC are:

- 42K of 32 bit words
- 1 random access disc
- a FORTRAN compiler (GRAFFIC conforms to ANSI FORTRAN)
- a FORTRAN overlay facility
- PLOT10 or other basic graphic software
- a storage tube with keyboard and cursor
- a hard copier (if hard-copy plots are required).

6. REFERENCES

1. Malin M R, Markatos N C, Tatchell D G and Spalding D B (1982): "Analysis and computation of Multi-Dimensional Coal Combustion Processes". ASME Paper No 82-FE-8, 1982

2. Markatos N C (1981): "Computer Simulation of Turbulent Fluid Flow in Chemical Reactors". Proc. 2nd World Congress of Chem. Engineering. Vol V, p 301.

3. Markatos N C and Mukerjee T (1981): "Three-dimensional Computer Analysis of Flow and Combustion in Automotive Internal Combustion Engines". Mathematics and Computers in Simulation, XXIII (1981), 354-366.

4. Markatos N C and Cox G (1982): "Turbulent Buoyant Heat Transfer in Enclosures Containing a Fire Source". Proc. of the 7th Int. Heat Transfer Conference, Vol 6, IH4, 373-379.

5. Rosten H I, Tatchell D G and Spalding D B (1983): "PHOENICS: A General Purpose Program for Fluid-Flow, Heat-Transfer and Chemical Reaction Processes". Paper to be presented at the 3rd International Conference on Engineering Software, London, April 1983.

6. Seppen J J (1982): "A Study of Flow Phenomena in Internal (piston) Combustion Engines". Ph.D Thesis, Technische Hogeschool Delft, Dept. of Mech. Eng., The Netherlands.

PHOENICS: A GENERAL-PURPOSE PROGRAM FOR FLUID-FLOW, HEAT-TRANSFER AND CHEMICAL-REACTION PROCESSES

H I Rosten, D B Spalding and D G Tatchell

Concentration Heat and Momentum Limited
40 High Street, Wimbledon, London SW19 5AU

SUMMARY

PHOENICS is a general-purpose computer program for predicting single or two-phase flow, heat-transfer, and chemical-reaction processes. Steady or unsteady phenomena in one, two or three space dimensions are handled. The governing equations for conservation of matter, momentum, energy and chemical species are solved by a finite-domain method. Full viscous effects are included; and the effects of turbulence are accounted for via a two-equation model involving the solution of transport equations for the kinetic energy of turbulence and its dissipation rate. Variable fluid properties are allowed for, and compressibility effects are included in subsonic, supersonic and transonic flows.

The program is structured in a way which separates the central core program which embodies the general-purpose computational procedures, from the user-accessible input portions. Two levels of user input are provided. Straightforward setting of data-input variables can be used for the prescription of a wide range of problems involving only activation of built-in facilities. For more complex problems provision is made in a controlled way for the attachment of specially-provided programming sequences or subroutines.

The paper discusses the mathematical basis, structure and organisation of PHOENICS and describes two typical applications.

INTRODUCTION

1.1 Background

The present paper describes the PHOENICS program, for simulating fluid-flow processes, and heat-transfer, chemical-reaction, and other related phenomena. The associated graphical-display program, GRAFFIC, is the subject of a companion paper, (Markatos and Pericleous, 1983).

The central part of the PHOENICS system is a single, general-purpose core program used for all applications, to which user access is provided at various levels to enable particular processes to be simulated. Identical versions of the core program are used by all users at all installations. This single-general-purpose-code strategy has been found to be beneficial in several ways: it facilitates maintenance and systematic development; it enables improvements to be made available immediately to all users; and it allows users to concentrate attention on modelling aspects of their particular problem, without being concerned with the numerical solution of the fundemental governing equations.

PHOENICS is currently in use at a number of installations for a wide range of problems, including nuclear-power, aerospace, turbomachinery, defence, marine, chemical-process, environmental and automotive applications.

1.2 The Present Paper

The intention in the present paper is to provide an outline description of PHOENICS and to illustrate its use by presenting recent examples of applications to practical problems.

Section 2 contains a discussion of the capabilities, structure and mathematical basis of the program, and a description of user-access facilities.

In Section 3 two recent applications are described briefly. These are: analysis of the response of a pressurised-water-reactor system to a loss-of-coolant accident; and the calculation of flow and combustion in a gas-turbine combustion chamber. Other PHOENICS results (including road-vehicle aerodynamics and a cooling-tower plume) are shown as examples by Markatos and Pericleous (1983).

Finally, Section 4 outlines current development plans and activities.

2. THE MAIN FEATURES OF PHOENICS

2.1 General Capabilities

PHOENICS is designed to simulate flow and heat transfer processes which are:

- one-, two- or three-dimensional;
- one-, or two-phase (in the latter case the phases may be dispersed as in a steam-water mixture, or separated by a distinct interface as for a bubble within a body of liquid);
- steady or transient;
- elliptic (ie fully recirculating) or parabolic (ie one-way, as in a boundary-layer flow);

- laminar or turbulent;
- subsonic, supersonic or transonic; and
- chemically reacting or inert.

2.2 Organisation

PHOENICS consists of three elements:

- The central 'EARTH' program, which is the common core program, used for all applications.
- A series of 'satellite' programs, each of which provides the specific, problem-defining input information which causes EARTH to simulate a particular process or item of equipment.
- A series of 'ground-stations', each associated with a particular satellite, each of which completes the problem-defining task.

The distinction between 'satellite' and 'ground-stations' is needed because the satellites are separate programs, which create data files to be read by EARTH, containing complete input specification. Because data transfer occurs once-for-all it is impossible to provide, through the satellite, information which varies during the calculation, or which involves interaction with the calculation procedure. The ground-stations are therefore provided to perform this role.

The PHOENICS system is shown diagramatically in Figure 1. The hexagonal satellites are shown communicating with EARTH through their own ground-stations (the 'blips' on the surface of EARTH).

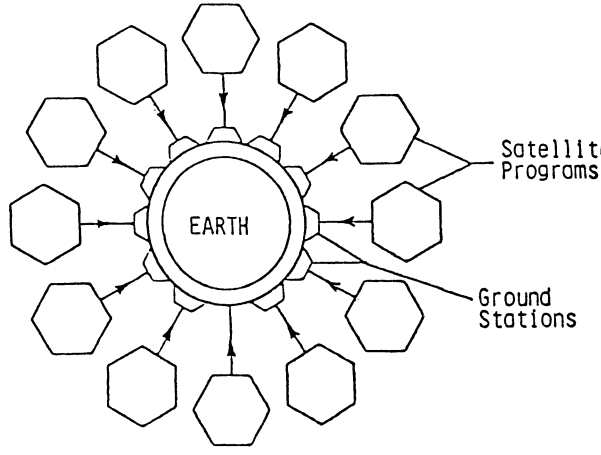


FIGURE 1: DIAGRAMMATIC REPRESENTATION OF THE PHOENICS SYSTEM

2.3 Mathematical Details of the EARTH Program

(a) Variables The independent variables are x, y and z if a cartesian coordinate system is used, or θ, r and z if polar coordinates are used. For unsteady flows, time, t, is also an independent variable.

There are twenty-three dependent variables which can be selected by the user:

- pressure p and the pressure correction p';
- three first-phase velocities, u_1, v_1, w_1;
- three second-phase velocities, u_2, v_2, w_2;
- phase volume fractions, r_1 and r_2, and a 'shadow' volume fraction, r_2^*, used particularly for two-phase problems involving chemical reaction;
- turbulence energy and dissipation rate, k and ε;
- first- and second-phase enthalpies, h_1 and h_2;
- enthalpies of a third fluid, h_3;
- four species concentrations, c_1, c_2, c_3, c_4;
- three radiation-flux sums, R_x, R_y, R_z.

The two quantities k and ε require some explanation. They represent local, time-averaged properties of the fluid turbulence, and are obtained from the modelled transport equations which constitute the two-equation, k$\sim\varepsilon$ model of turbulence (Launder and Spalding, 1974).

(b) Governing Equations The equations which express the physical laws of 'conservation' of mass, momentum, energy, species and turbulence properties for each phase have the common form:

$$\frac{\partial}{\partial t}(r\rho\phi) + \text{div}\,(r\rho\vec{V}_\phi - r\Gamma_\phi \text{grad}\phi) = rS_\phi$$

where: r $\equiv$ phase volume fraction; $\rho \equiv$ density;

ϕ $\equiv$ any of the dependent variables;

$\vec{V}$ $\equiv$ velocity vector;

Γ_ϕ $\equiv$ exchange coefficient (laminar or turbulent); and

S_ϕ $\equiv$ source or sink term.

Algebraic 'finite-domain' equations are derived by integration of the differential equations over grid cells of a rectangular or cylindrical grid. The general form is:

$$\phi_P = \frac{\Sigma_i \, a_i \, \phi_i + b}{\Sigma_i \, a_i - c}$$

where ϕ is the dependent variable, ϕ_P is the value at the grid cell in question, and Σ_i denotes summation over the neighbouring grid points (of which there will be six in a three-dimensional case, four in a two-dimensional case, and two in a one-dimensional case).

The coefficients a_i represent the effects of convection and diffusion, and $(b+c\phi_P)$ contains the transient term and the integrated source term for the cell (including, for the momentum equations, the pressure-gradient term). The use of upwind differencing in evaluating the convection terms ensures that the a_i coefficients are always positive, as required for numerical stability.

(c) Solution Methods The finite-domain equations are solved using an iterative guess-and-correct procedure which is fully-implicit in transient cases, and based on a variant of the SIMPLE algorithm (Patankar and Spalding, 1972, and Spalding 1980a). For two-phase-flow calculations the IPSA alorithm (Spalding, 1980b and 1981) is used.

Solution is obtained by repeated z-direction sweeps through the integration domain. The whole set of cells is regarded as consisting of one-cell-thick 'slabs', extending in the x and y directions and piled one on top of the other in the z-direction.

A single sweep therefore starts with attention being paid to the bottom (low-z) slab of cells. The finite-domain equations are solved for all the cells in this slab, the values of ϕ's at the next-higher slab being regarded as known. Attention then passes to the second-slab, the ϕ-values there being adjusted by reference to those in the slabs both above and below. Then the next-higher slab is attended to; and so on, until the adjustment sweep has been completed.

At the end of a sweep, of course, the final solution has not yet been achieved, so the process must be repeated until a reasonable degree of convergence is obtained.

When the flow is a transient one, the same iterative-sweep procedure is employed. However, the number of sweeps needed to achieve convergence at a particular time-step reduces greatly with the magnitude of the time interval; when this is very small, a single sweep is all that is required.

As well as the full, elliptic, iterative solution procedure, EARTH contains as a user-activated option a marching-integration, parabolic solution procedure. This arranges that only a single sweep is performed, and that values at each slab are overwritten as the solution proceeds downstream. The program therefore operates in the same way as a special-purpose parabolic or boundary-layer procedure, and automatically effects the usual storage reduction.

(d) Non-regular Geometries The usual technique for representing non-regular geometries in PHOENICS involves wholly or partially blanking-off grid cells via 'porosities'. Four porosities are defined for each grid cell; area porosities for cell faces normal to each of the three coordinate directions, and a volume porosity. Each represents the proportion of the normal area or volume which is not occupied by solid.

This technique can be used to represent non-regular flow-domain boundaries (which can thereby be allowed to cross the regular rectangular or cylindrical grid in an arbitrary way), and both porous and impervious internal structures.

The porosity arrays are input by the user, in the satellite program. The blockages are then automatically taken account of in the finite-domain equations solved by EARTH.

An alternative method of handling non-regular geometries involves the use of non-orthogonal body-fitted coordinates, in which the grid mesh is distorted to fit the physical boundaries of the problem. A system of this kind is currently being introduced into EARTH, to be activated by specification of domain-boundary coordinates in the satellite program. This would generally be used for smoothly-varying geometries, such as turbine blades, or ship or submarine hulls.

(e) Storage Allocation Array dimensions and computer-storage allocations are set internally by EARTH, according to the numbers of grid nodes input by the user, the dimensionality of the problem, the variables to be solved for, whether the problem is steady or unsteady, whether it is parabolic, and so on.

The program then tests the total required variable storage against that available in-core on the machine in question (which information must be provided by the user). If insufficient core is available, the program automatically selects one or more of a range of out-of-core storage options.

These arrange that selected three-dimensional arrays are stored on disc, and that the required information for the slab being solved for and its two neighbours is transferred into and out of in-core storage, as the solution proceeds

from slab-to-slab. In the full out-of-core mode, no three-dimensional arrays are stored in core. This enables very large three-dimensional grids to be used within limited computer-storage constraints.

2.4. User Access to PHOENICS

(a) The Satellite Program The satellite program comprises a main program, and a BLOCK DATA subroutine; it may also, if required, include subroutines FLDDAT, SPCDAT and PORDAT. The functions of each of these will now be explained.

BLOCK DATA contains all the main variable settings which serve to define a problem, the main ones being:

- The flow type (ie parabolic?, cartesian or polar?, steady or transient?).
- Overall dimensions of the flow domain.
- Numbers of grid nodes and time-steps, and their distributions. If the number of nodes in one direction (say x) is set to unity, the program then automatically treats the flow as two-dimensional (in yz).
- The dependent variables to be solved for. If second-phase variables are selected, the full IPSA two-phase-slip solution algorithm is automatically activated.
- The fluid properties. There are three options: the property may be constant, in which case, the value is specified; or built-in formulae (such as the ideal gas law for density, or the two-equation turbulence model for effective voscosity) may be selected by setting an index; or special sequences to be provided by the user in the ground station can be activated by setting the same index to -1.
- The formulae to be used for interphase transport processes. Again, simple built-in formulae are selected by setting an index; alternatively, appropriate user-provided sequences in the ground station can be activated.
- The initial fields. These may be uniform, in which case the value for each variable is input here, or non-uniform, in which case the call to subroutine FLDDAT is activated.
- The boundary conditions. A range of basic options is provided in BLOCK DATA, which allow regions of the flow domain to be defined over which simple boundary conditions can be specified, including: fixed mass inflow or outflow; fixed pressure; fixed or linear source or sink of any variable. In addition, built-in turbulent wall functions can be activated. Other, more-complex, boundary-condition information (involving, for example, problem-specific correlations for heat-transfer coefficient) would be provided in the ground-station.

- Numbers of iterations and sweeps.
- Relaxation factors and convergence criteria.
- Frequency and type of printout.

The intention is that the input facilities provided by BLOCK DATA should be as comprehensive as possible, so that, as far as is practicable, the user need concern himself only with this subroutine. User actions are required elsewhere only when the input information is too complex to be defined in terms of a limited number of parameters (as, for example, when arbitrary, non-uniform initial fields are required for specific variables, in which case FLDDAT is used), or when interaction with the solution procedure is necessary, in which case the ground station is used.

The primary function of the main program of the satellite is simply to write the data file containing the variables specified in BLOCK DATA, to be read by EARTH. However, it also contains provision, if required, to modify any of these data, and to transmit up to 30 data files having made modifications between each transmission, enabling parametric studies to be made easily.

Subroutine PORDAT is called only if activated by the appropriate logical variable in BLOCK DATA. Within it the user prescribes the porosity arrays used to represent blocked-off regions of the grid, as described earlier. These arrays are then stored as a data file, to be read by EARTH.

Subroutine FLDDAT is where the user specifies non-uniform initial fields for variables identified in BLOCK DATA. These are again stored as a data file, which is automatically read and acted on by EARTH.

In subroutine SPCDAT, the user can provide any special data required in his own ground-station sequences.

(b) The Ground-Station The basic ground-station is a single subroutine, GROUND; other subroutines, called from GROUND, can of course, be added by the user.

GROUND is called from EARTH at all stages in calculation at which the user might wish to take action. In particular, it is called at the start and end of each time-step, sweep, slab iteration, and hydrodynamic loop, and when the source term for each variable is computed. Special calls are also provided (activated by index settings in BLOCK DATA), to calculate densities, viscosities, and interphase transport coefficients.

Actions taken by the user in GROUND might include:

- Modifying any of the input variables from BLOCK DATA, to represent, for example, time-varying boundary conditions.
- Providing appropriate formulae for fluid properties, or attaching existing programs or subroutines for (for example) steam-water properties.
- Providing appropriate source terms for particular variables, to correspond to special boundary conditions, heat sources or sinks, etc.
- Inserting or modifying empirical formulae for interphase and wall friction and heat transfer.
- Performing calculations using computed variables to deduce special derived quantities (eg total force or heat transfer rate at a particular portion of the domain boundary), or to check the solution for internal consistency.
- Providing special printout.

In order to enable communication to be made with EARTH without the user understanding the internal array referencing system (which is designed for computational efficiency, rather than transparency), special 'translating' subroutines are provided. These allow the user to access variables in EARTH and store them in local arrays in GROUND, to fix variables in EARTH, to modify source terms, and to print arrays, all according to defined rules which do not require knowledge of the internals of EARTH.

3. TWO TYPICAL APPLICATIONS OF PHOENICS

3.1. Loss-of-Coolant Accident in a Pressurised-Water Reactor

A matter of concern in the design of pressurised-water-reactor (PWR) systems, is the possibility of a loss-of-coolant accident (LOCA), in which a break in the primary coolant circuit leads to depressurisation, boiling of the coolant, and reduced cooling of the reactor core.

The coolant system of a PWR generally comprises four separate circuits for each reactor. For certain accident conditions (principally when the break occurs in the reactor itself), each circuit will essentially behave in the same way. Computations need then be performed for only one equivalent circuit.

The coolant circuit considered by PHOENICS is shown diagram-atically in Figure 2. Under normal operation, the pressurised water is circulated anticlockwise, upwards through the reactor, through the tubes of the steam generator, and then, via the downcomer back to the reactor. The accident

considered here involves a 4inch diameter 'small break' in the lower plenum at the base of the reactor.

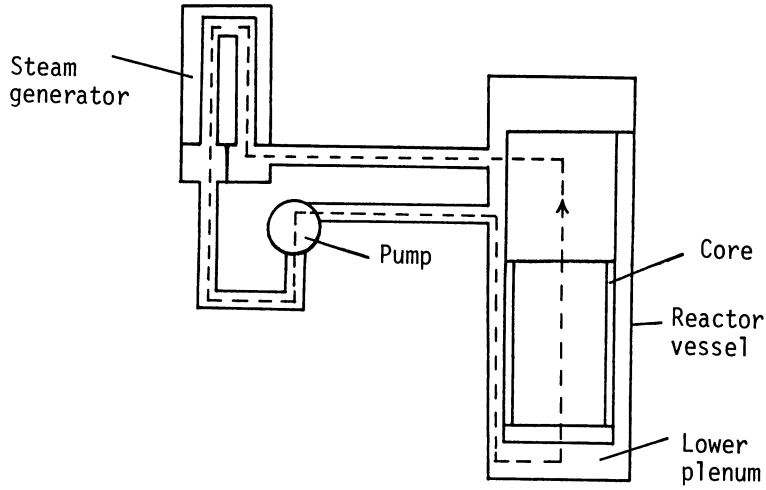


FIGURE 2: SCHEMATIC DIAGRAM OF THE PWR COOLANT CIRCUIT CONSIDERED

The problem is transient, and two-phase, and the circuit is represented one dimensionally. The PHOENICS variables solved for are p, p', u_1, u_2, r_1, r_2, h_1, h_2. In addition a separate calculation sequence has been provided (in subroutine GROUND) for the temperature of the reactor fuel rods.

For interphase friction and heat transfer, various correlation sets are being investigated, including those used in the TRAC-P1 program (Reference 11). These, and other special features of the problem (eg appropriate steam and water property formulae, pressure drop formulae, momentum source terms to represent the pump, etc) have also been provided in GROUND.

The calculation starts from computed, steady-state conditions for normal operation. The transient is initiated by the introduction of the break as an appropriate pressure-dependent outflow rate. The reactor-control event sequence is then imposed on the calculation as time-varying boundary conditions in, for example, heat input from the core (reduced to 3% between 16s and 100s) and pump torque (reduced to zero between

26s and 35s).

Figure 3 shows, for a typical calculation, variations of liquid content around the circuit (the shaded portion) at particular times in the transient. Also shown are flow directions for liquid (solid arrows) and gas (dashed arrows) in parts of the circuit.

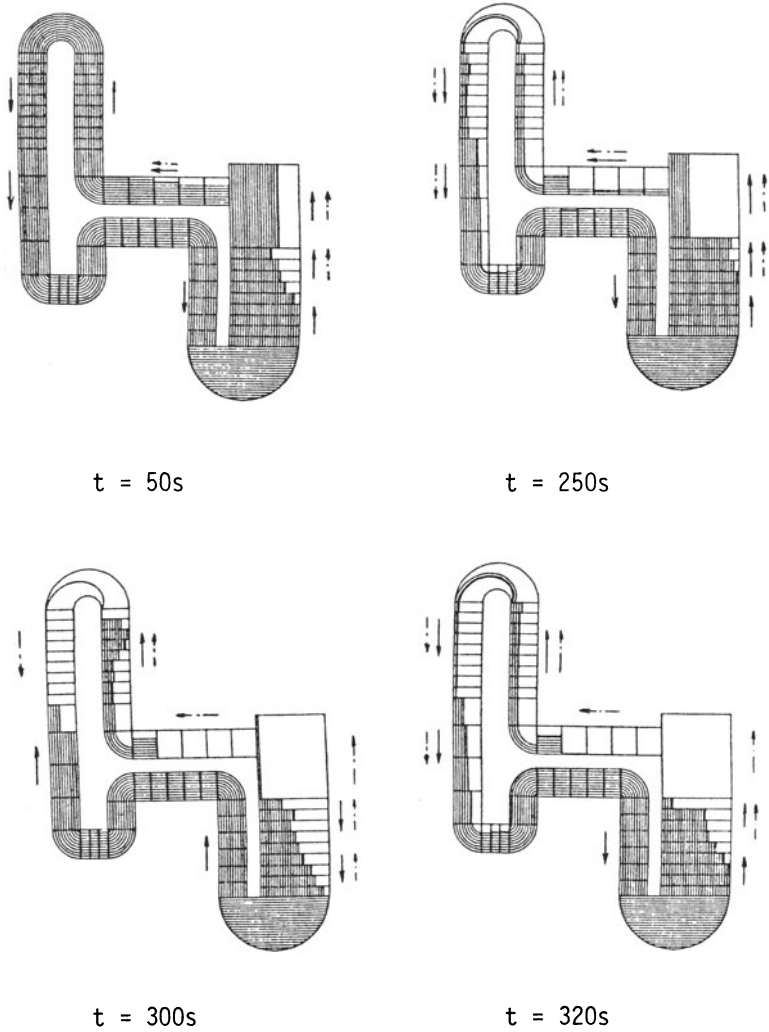


FIGURE 3: TYPICAL PWR LOSS-OF-COOLANT ACCIDENT RESULTS: CIRCUIT LIQUID CONTENT AND FLOW DIRECTIONS

In the early stages of the transient, at 50s, the depressurisation of the system caused by the break, and the reduced coolant flow as the pump runs down, lead to boiling in the upper part of the reactor core. This vapour is carried into the upper part of the reactor vessel and the adjacent pipework.

At 250s (as at 50s) the flow continues to circulate anti-clockwise. The upper part of the circuit, including the steam generator, is now largely filled with vapour, while the liquid in the core continues to be replenished from the lower part of the circuit.

Following this, at 300s, the liquid flow in the core and the lower parts of the circuit has evidently reversed, leading to draining of the core. This process then itself reverses, so that at 320s the core is refilling again from below.

These results serve to illustrate certain aspects of the calculations currently being performed. It must however be emphasised that, within the context of the present paper, it has been possible only to show a small part of the results of the reported calculation, and, furthermore, that work is continuing to investigate the effects of various idealisations and empirical representations in the model. A more complete description of this work is given by Markatos et al (1983), and the associated coding used to set-up the problem in PHOENICS is described by Markatos and Rawnsley (1982).

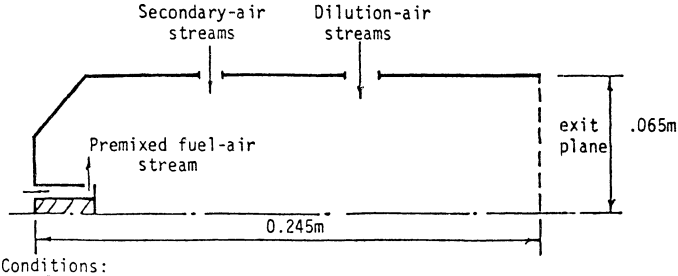
3.2 Gas-Turbine Combustion Chamber

The problem considered is a typical gas-turbine can combustor, as illustrated in Figure 4. The chamber is cylindrical, with a conical end. Premixed fuel (presumed vapourised) and air enter from the mixing chamber at the left-hand end. Additional air is injected radially from the two rows of ports in the outer casing. The first row comprises ten ports, and the second row five ports, each equally spaced around the circumference of the chamber.

The problem is three-dimensional (as a consequence of the discrete injection ports), steady and single phase. In the circumferential direction, symmetry permits the calculation to be restricted to 36^{o} - ie to include one secondary port and one-half of a dilution port.

The combustion model used is of the kind described by Spalding (1971), in which a one step reaction is presumed of the form fuel + oxidant → products. The local concentrations of fuel, oxidant (ie air) and products are obtained from two solved-for concentrations representing the 'mixture fraction' f and the concentration of fuel, m_{fu}. The turbulent reaction rate (as given by Spalding, 1971) appears as the (negative) source term of m_{fu}.

The variables solved for are p, p', u_1, v_1, w_1, k, ε, h_1, c_1 and c_2. The concentrations, c_1 and c_2, represent f and m_{fu}; the reaction-rate source term for the latter is programmed in GROUND.



Inlet Conditions:

Stream	Fuel-Air	Secondary	Dilution
Flow rate kg/s	0.118	0.016/port	0.029/port
Radial velocity m/s	100	-40	-40
Temperature ^{o}K	773	773	773
Fuel/air ratio	1:17.24 (stoichiometric)	0	0

FIGURE 4: THE COMBUSTION-CHAMBER CONFIGURATION CONSIDERED

Figures 5 and 6 show typical computed results, at planes $\theta = 0$ (containing both secondary and dilution ports) and $\theta = 36^{o}$ (containing a secondary port only).

Figure 5 shows the predicted velocity vectors. Clearly evident are the high radial velocities induced by the incoming fuel-and-air stream, and the consequent negative axial velocities just to the right of the inlet as the fluid is entrained into the radial inlet jet. Further downstream the effects of the injected streams, and the asymmetry caused by the dilution-air injection, can be seen.

Figure 6 shows the temperature fields. The highest temperatures, caused by combustion of the incoming gases, occur in the slow-moving region behind the inlet. The cooling effect of the injected air is evident further downstream.

A more complete presentation of the results and description of the problem set-up for this case are given by Spalding and Tatchell (1983), and the input coding is described by Malin et al (1982).

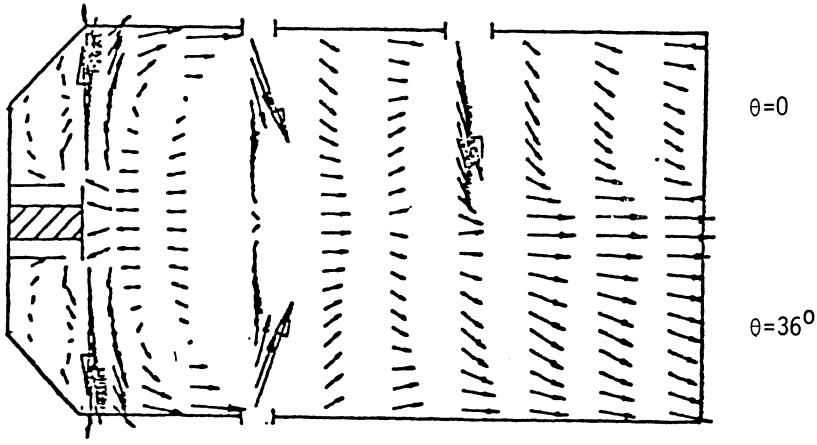
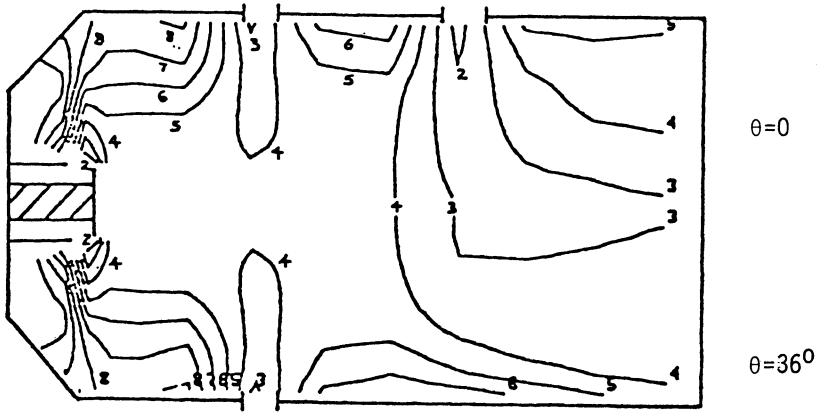


FIGURE 5: COMBUSTION CHAMBER: VELOCITY VECTORS AT θ= 0 and 36°



Contours equally spaced between:
1: 773°K 11: 2400°K

FIGURE 6: COMBUSTION CHAMBER: TEMPERATURES AT θ = 0 and 36°

3.3. Computer Requirements

All calculations have been performed on a Perkin Elmer 3220 32-bit word mini-computer. The computer times given below are for this machine; large main-frame computers would be between ten and one hundred times faster.

For the 54-node grid used for the LOCA calculation reported in Section 3.1, a typical time step with 20 iterations requires about 0.5min of cpu time. For the complete 450s calculation, with time steps of 0.5s, the total run time was therefore 450 minutes. The run was made within about 300kB of core.

For the combustor calculation described in Section 3.2, a grid of 6 x 10 x 13 (θ x r x z) was used, and 80 iteration sweeps were required. The total cpu time was 18 minutes, and the storage requirement (with all variables in-core) was 375kB.

4. CLOSING REMARKS

In addition to the applications just described, PHOENICS is currently being applied to a wide range of problems including fast-breeder-reactor thermal-hydraulics, gun ballistics, external hydrodynamics of ships and submarines, external aerodynamics of road vehicles, power-station condensers, and smoke movement in enclosures. As well as at CHAM, the program is being used at a number of in-house commercial and university installations, and via the major computer bureaux networks. The mainframe and mini-computers on which it is operating include Cray, Cyber, Control Data, IBM, Amdahl, Univac, Prime and DEC equipment.

A major priority in the development of the EARTH program is the introduction of a general body-fitted coordinate system for the representation of streamlined geometries such as gas-turbine blades and vehicle bodies. Other current development activities include the introduction of improvements in computational efficiency, in respect of speed and storage, and extending the range of built-in models, property formulae and empirical correlations.

Attention is also being given to input and output facilities. As an alternative means of providing input data, an interactive program is in preparation which, by a simple sequence of questions and user responses, prepares the complete input data set normally created by the satellite.

REFERENCES

1. Launder B E and Spalding D B (1974)
'The Numerical Computation of Turbulent Flows.'
Computer Methods in Applied Mechanics and Engineering
3, pp 269-286.

2. Malin M R, Tatchell D G and Wang Ying-Shi (1982)
'Calculations of Steady, Three-Dimensional, Turbulent,
Reacting Flow in a Combustion Chamber'.
CHAM Limited Report PDR/CHAM UK/19.

3. Markatos N C and Pericleous C A (1983)
'GRAFFIC: A Computer Package for the Interactive Graphical
Representation of Fluid-Flow Phenomena'.
Paper submitted for presentation at Third International
Conference on Engineering Software, Imperial College,
London, 1983.

4. Markatos N C and Rawnsley S M (1982)
'Analysis of Small-Break Loss-of-Coolant-Accident in a
Pressurised Water Reactor'.
CHAM Limited Report PDR/CHAM UK/11.

5. Markatos N C, Rawnsley S M and Tatchell D G (1983)
'Analysis of a Small-Break Loss-of-Coolant Accident
in a Pressurised Water Reactor'.
Paper submitted for presentation at Conference on Heat
and Fluid-Flow in Nuclear and Process Plant Safety.
Institute of Mechanical Engineering, London, 1983.

6. Patankar V and Spalding D B (1972)
'A Calculation Procedure for Heat, Mass and Momentum
Transfer in Three-Dimensional Parabolic Flows'.
Int. Jnl. of Heat and Mass Transfer, Vol. 15,
pp 1787 - 1806.

7. Spalding D B (1971)
'Mixing and Chemical Reaction in Steady Confined
Turbulent Flames'.
13th Int. Symposium on Combustion, The Combustion
Institute, Pittsburgh.

8. Spalding D B (1980)
'Mathematical Modelling of Fluid Mechanics, Heat
Transfer and Mass Transfer Processes; a Lecture Course'.
Imperial College, Mech. Eng. Dept. Report HTS/80/1.

9. Spalding D B (1980)
'Numerical Computation of Multi-Phase Fluid Flow and Heat-Transfer'.
Recent Advances in Numerical Methods in Fluids, Vol. 1.
Edited by C Taylor and K Morgan, Pineridge Press, Swansea.

10. Spalding D B (1981)
'Developments in the Procedure for Numerical Computation of Multi-Phase-Flow Phenomena with Interphase Slip, Unequal Temperature, etc.
2nd Int. Conference on Numerical Methods, Maryland University September 1981.

11. Spalding D B and Tatchell D G (1983)
'A General-Purpose Fluid-Mechanics Computer Program For Turbomachinery Applications'.
Paper submitted for presentation at CIMAC Congress, Paris, June 1983.

12. "TRAC-P1: An Advanced Best-Estimate Computer Program for PWR LOCA Analysis. Vol. I: Methods, Models, User Information, and Programming Details', Los Alamos Scientific Laboratory Report LA-7279-MS, Vol I (NUREG/CR-0063). June 1978.

COMPUTER AIDED PIPING SYSTEM AT ENGINEERS INDIA

S. Krishnamurti, G.M. Deshpande and V.K. Gupta

Systems Engneering and Computer Services Division
Engineers India Limited

1.0 INTRODUCTION

Engineers India (EI), primarily an engineering consultancy organisation, has been operating in the fields of Refineries, Petrochemicals, Fertilizers, Non-ferrous Metallurgy, Offshore Engineering, Heat & Mass Transfer Equipment and other allied areas. EI offers comprehensive services from concept to commissioning covering various phases such as feasibility, process design, detailed engineering, procurement, construction, commissioning and project management in its various areas of operation. Systems Engineering and Computer Services division of EI has a comprehensive library of over 300 well documented computer packages covering the above areas.

EI's principle area of operation is chemical process plants wherein piping system constitutes one of the important segments. Piping design finds its place somewhat late in a typical time schedule of a chemical project. This is because the realistic planning for piping cannot be made unless the process design is frozen and building and equipment layouts are finalised to a great extent.

The success of the computer based system depends on its integration into an organisation's method of operation. This certainly does not emphasize that mere computerisation of existing manual procedures

would prove most successful. Even though it may find immediate acceptance, it may not prove efficient. Improving the working environment by reducing the monotony may improve the system's efficiency more than most of the efficiently written programs. More emphasis is therefore given in this paper on describing the piping system, as well as the program and the approach.

Computer Aided Piping System (CAPS) can be best described as a man/machine interface and aims at design, drafting and monitoring of piping from the stage of conception till mechanical completion at construction site. Man provides the 'design or thought process' whereas the computer renders whatever assistance that is necessary in carrying out this process. The computer acts as an extension of engineering skill to increase the producitivity. Graphical display devices are used wherever part of the design activity or results of analysis need to be visualised.

CAPS aims at producing isometric drawings, consolidating material take-offs and tracking the individual items through material control till they get positioned at the construction site. Functional demarkation between these sub-systems being quite distinct, EI opted to develop these systems independently. The individual systems were developed and stabilized over a period of time and subsequently the task of integration of these sub-systems was undertaken and a well integrated piping system has been evolved.

The benefits and advantages that can be conceived in introducing CAPS in an engineering consultancy organisation such as EI can be broadly outlined as below:

- Analysis of more alternatives aiding in stringent evaluation of critical items.

- Visual display possibilities enhance checking of fouling in isometric drawings.

- Integrity of material take-offs.

- Preparation of a Material Requisitions.

- Aid in timely placement of purchase orders.

- Follow-up of material inspection as per purchase order schedule.

- Preparation of cost summaries.

- Management of project by exception.

- Setting up of integrated data bases.

CAPS has been given due importance at EI, having realised that such a system would expedite and considerably aid in early completion of chemical plants and would ideally suit a consultancy environment such as EI, aiding the designers, draftsmen, fabricators and the management at various stages of project execution.

2.0 THE SYSTEM

CAPS encompasses activities of piping from the stage of generating isometric drawings till the issue of piping materials at site to the Contractors. Overview of CAPS is shown at Figure-1.0, indicating all the three sub-systems namely, Isometric, Material Take-Off and Material Control. The system is well integrated in the sense that the output of Isometric Phase is accepted as input in the Material Take-Off Phase and output of Material Take-Off Phase is accepted as input in the Material Control Phase. In addition some of the standard data bases are shared by all the three sub systems.

2.1 Isometric Phase

A 3 dimensional view of piping layout when represented in 2D in a typical manner is termed as piping isometric. System flow for isometric phase has been described at Figure-2.0. The input is classified in two groups viz. Permanent input in the form of standard data base files and variable input configuring individual isometric. The data base files hold information on dimension data of about 100 individual piping components such as pipes, flanges, valves and fittings, etc. Data of about 200 frequently used combinations of basic piping elements termed as assemblies is also stored as standard data. The symbol of each piping element called as Menu, is stored in the 39 Menu libraries holding data for about 1000 Menus. The variable input consists of line specifications, X, Y, Z co-ordinates as available on General

Arrangement Drawings (GADs), of each item that needs to be located and that of junction points where there is a change of direction and item codes of piping components.

This data input is validated for possible logical errors such as undefined branches, line retracing, invalid loops and branches, etc. The validated data is then processed for picking up the face to face dimensions of standard piping components based on line specifications. The representation of items for four different end preparations, viz., flanged, butt welded, socket welded and screwed ends being different this is established at this stage for selection of proper symbols and relevant menus are picked up from the respective menu libraries. After restoring the information on standard dimensions, analysis is done for finding out the current and next segment orientation of each component which also decides whether the segment is skew or non-skew. The pipe cut off lengths are computed thereafter. Having known these details the drawing co-ordinates for each component are established and analysis for intersection is performed. Depending upon the complexity of each drawing micro level analysis is performed for expanding or contracting the pipe lengths so as to improve the asthetics and clarity. Once the analysis for drawing the main body of the isometric is completed, the data is further analysed for dimensioning, which again is user independent. In this module the necessary records needed for dimensioning are identified and soft zones are generated for each segment. These are then analysed for intersection. In the event of overlapping, the soft zones are broken into sub-segments and the analysis is repeated for selecting the best side of the sub-segment for dimensioning. Thus fouling is minimized and clarity of the drawing is improved considerably.

Isometric drawing is then spooled for plotting on the plotter. Optionally the isometric can be displayed on the graphic terminal for carrying out interactive updation, if desired.

Data for Material Take-Offs containing the details such as item codes, quantity, item specification, etc. are generated from the isometric input and line wise material take-offs are taken on printer. A sample isometric output is shown at Figure-3.0.

2.2 Material Take-Off Phase

The global data base holds EI's standard piping specifications and is accessed in this phase. Project specific changes in the piping specifications are also accepted. This phase generates linewise bill of materials, zone/plant summary, item break-up and line index. Escallations based on standard escallation data base are provided. A unique seven digit code generated by the computer at this stage for new items is used in all subsequent transactions. Material Requisitions are generated alongwith information required for the next phase, namely, Material Control.

2.3 Material Control Phase

Planning and control of materials constitutes an important segment of project management activities. It enables timely availability of right materials in right quantities and therefore helps to significantly cut down the project completion time and material costs. Material control phase of CAPS monitors progress of various activities of bulk materials such as pipes, flanges, fittings, valves, gaskets, bolts and other miscellaneous items. This involves interaction between various functional groups in the project like, Engineering, Procurement Services, Inspection and Expediting, Stores and Construction either directly amongst each other or through Project Services Division as an interface. A project of an average capacity involves about 6000 to 8000 different piping items each to be procured 4 to 6 times during the project duration and each time in varying quantities. Different activities that have to be carried out in succession on each single item are identifying the requirement through MTO, requisitioning, ordering, inspection at vendor's shop, receipt of material at site, issue of material for fabrication and return of balance material. The total number of such transactions run into millions and therefore efficient monitoring becomes essential. A seven digit item code internally generated is referred in all the documents. The reporting is based on the principle of exception in that whereas all items are monitored continuously only the exceptions are reported. Status information also can be had on requests. The status reports constitute material status, procurement status, material allocation to construction and unplanned issue status. Exception

reports are generated on requisitioning, ordering, release and receipts.

CAPS has extensive application at construction site while clearing the pipelines for fabrication and erection in regard to material allocation from the available stock giving due weightages to system priorities. CAPS additionally monitors complete status of unplanned issues at construction site which is of immense help to the project division for replenishment of these items.

2.4 Extensions

CAPS has extensions to the cost monitoring system and helps in providing material cost summary for the Project.

Efforts are presently underway to integrate the existing piping flexibility analysis package with CAPS.

3.0 EI'S EXPERIENCE IN IMPLEMENTING CAPS

Having developed CAPS in an integrated manner, EI initiated well laid down plans for implementation of CAPS. The major task of CAPS specialists in implementing the system was of educating the potentialities of such a scheme to a group of technologists/engineers, draftsmen, fabricators and erectors who constitute the user group. Fortunately EI's environment is well oriented towards computerisation. Initially CAPS was introduced on one major project of EI. The concerned users along with group leaders of other projects were imparted training through an introductory course on CAPS so as to have exposure to the system. Subsequently, on-the-job training was given to the concerned project personnel for a comparatively longer duration. Once the user independence was achieved, support services of CAPS specialists were made available to the user department on need basis. Successful implementation of CAPS on one project gave confidence and impetus for usage of CAPS on other projects. All major projects handled by EI are presently getting benefitted by the usage of CAPS.

Efficient usage of CAPS on a typical project results in substantial saving of manhours. The study shows that on a project of an average size having about 1500 isometric lines and 3 material

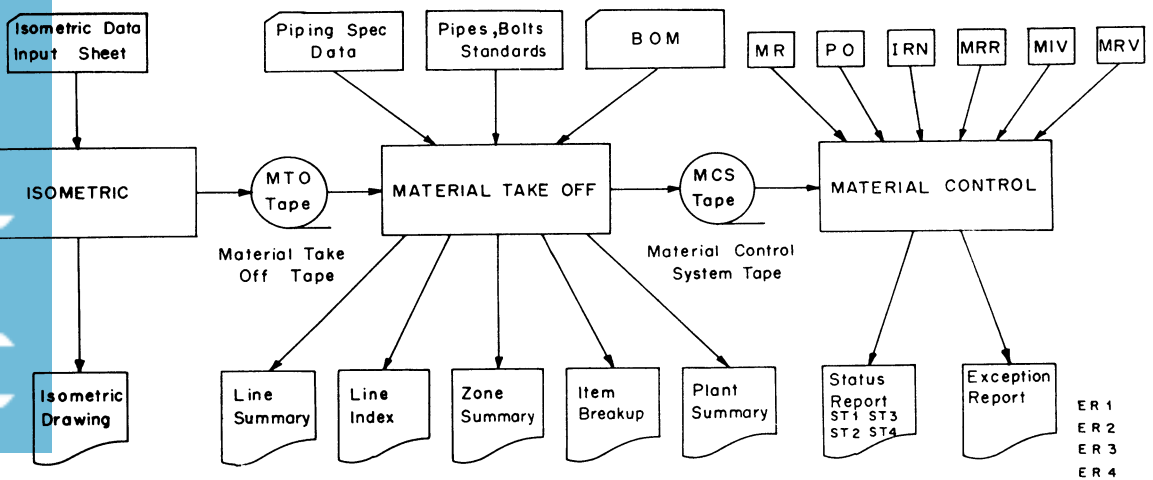
take-offs during the entire project duration of about 2-1/2 to 3 years; utilisation of CAPS helps in crashing the project duration by a span of about 2 to 4 months and achieves substantial savings in manhours. In addition, large number of reports are generated which improve the quality of work performed and help in documenting the project proceedings in the most professional manner.

4.0 CONCLUSION

Introduction of CAPS gives EI an edge over other consulting organisations in the competitive international market. Usage of CAPS has led to better utilisation of human resources apart from the various tangible and intangible benefits already outlined. In the opinion of the authors for large engineering projects, to maintain stringent schedules, integrated systems like CAPS covering the entire span of activities and monitoring them on the principle of management by exception suitably inbuilt in the system are essential.

5.0 ACKNOWLEDGEMENT

The authors wish to thank EI's management for having given the encouragement and providing the necessary infrastructure for preparing the paper.



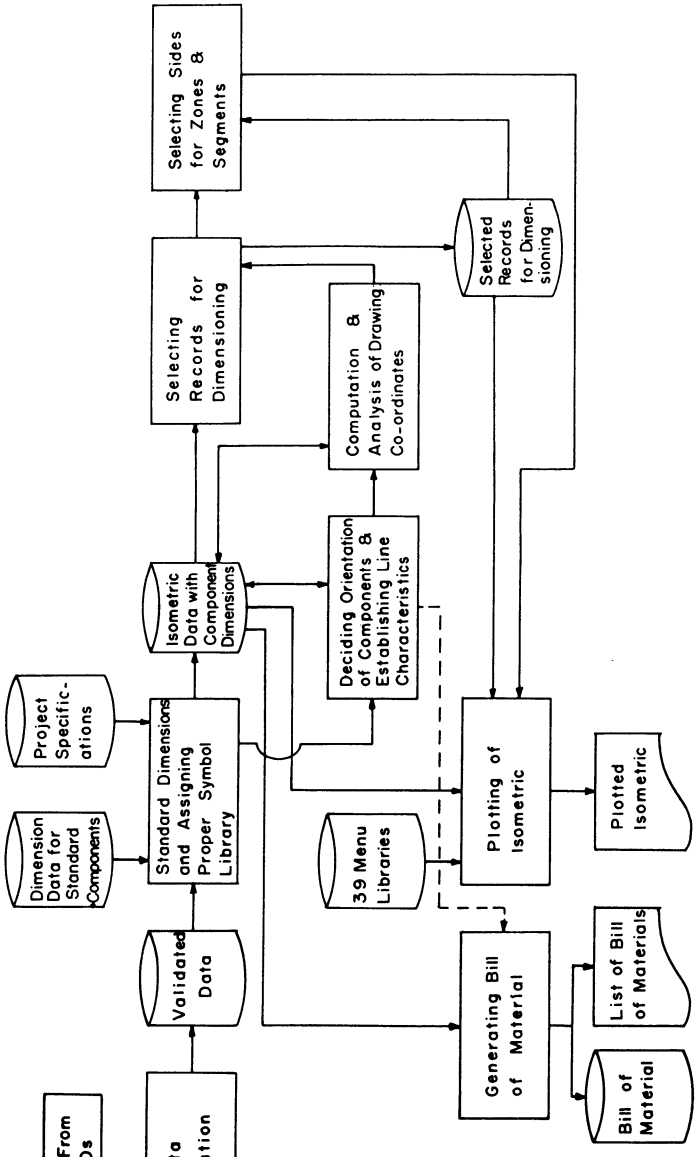
LEGEND

BOM	Bill of Materials	ST 1	Status Report on Item	ST 3	Material Allocation Report
MR	Material Requisition	ST 2	Status Report for Procurement	ST 4	Unplanned Issue Status Report
PO	Purchase Order	ER 1	Exception on Requisition		
IRN	Inspection Release Note	ER 2	Exception on Ordering		
MRR	Material Receiving Report	ER 3	Exception on Release		
MIV	Material Issue Voucher	ER 4	Exception on Receipt		
MRV	Material Return Voucher				

OVER VIEW

COMPUTER AIDED PIPING SYSTEM

FIGURE - 1·0



SYSTEM VIEW OF ISOMETRIC PHASE

FIGURE-2·0

10 COLUMN.H-17
20 C.L.E-506
25 NOZZLE-N4
100 50FY302-C.V.
140 T-SUPPORT
160 10-SH-50-821-1-A2A
170 2-SH-50-533-1-B3A
195 3/4DBF
PROJECT: LPG-II
JOB NO: 1891
LN:SH-50-524-1-8 SP:B3A
200 3"DRIP LEG
STRESS RELIEVE RADGRPH
NO 0002
INSULATION SIZE TYPE THK
8.00 IH 95
TEST PR.36.00 MED.WATER
DES PR./TEMP 21.00 KG/SQ CM 200.00 C
REF DRG NOS: 506 538
P&I NO: 503
M/S.KRIRSBS INDIA LIMITED NEW DELHI

FIGURE -- 3.0

HIGH-SPEED CONTROL AND DATA ACQUISITION SYSTEM FOR A NOVEL COMBUSTION RIG

D.J. Barker*, P.H. Prest* and J.P. Packer+

* Electronics & Instrumentation Section +Thermal Power Group, School of Engineering, University of Bath.

SYNOPSIS

High-speed data acquisition and control of a combustion rig using a microcomputer. The system hardware and software are described and the method of operation detailed. Typical experimental results obtained are presented.

INTRODUCTION

This paper describes a high-speed data acquisition and control system as applied to a novel combustion rig. The combustion rig reproduces the conditions of high air pressure (up to 60 bar), high temperature (up to 850k) and swirling air motion to be found at the start of fuel injection in small to medium sized direct-injection diesel engines. A single-shot fuel injection system supplies fuel to the chamber under engine-like conditions. The mechanical details of the rig are more fully described by Packer (1983) and Packer, Wallace et al (1983).

The fuel injection and combustion event typically occupies from 1 to 30 m-secs and the maximum amount of information from the experiment is required during this period. High-speed cine photography (up to 8000 frames per second) is used to record the fuel spray and flame patterns. Fuel injection pressure, injector needle valve opening and combustion chamber pressure are monitored by various transducers. The microcomputer-based data acquisition and control system controls the fuel injection sequence and records the subsequent transducer outputs at sampling rates up to 70 khz.

At the end of the test the transducer outputs captured by the system may be examined on an oscilloscope. If the data is satisfactory it may then be transferred by another micro-computer to the University mainframe computer for complete analysis.

The data acquisition system achieves a maximum sampling rate of 70 KHz by making use of programmable counter-timer interface chips to control the number of samples, sampling rate and delay times for experimental events and start of data collection. Control of the data acquisition cycle therefore occurs automatically, allowing the processor to collect and transfer to ram faster than if it had to carry out these functions itself. This gives a fast flexible system and permits software variation of the system parameters. It is relatively inexpensive and may be suitable for a wide range of experimental applications.

DESCRIPTION OF THE SYSTEM AND ITS METHOD OF OPERATION

The system consists of a microcomputer (Commodore PET) and three major cards viz

a) the data acquisition card
b) the oscilloscope output card
c) the solenoid control card

and is shown schematically in figure 1. The solenoid card is used to control a single shot of fuel into the combustion chamber. The single shot is achieved by running a fuel injector pump with its throttle normally closed but with a high-power solenoid attached to the throttle linkage (the fuel rack). To produce a single injection the solenoid is energised for a sufficient duration to open the fuel rack for one pumping cycle.

It is assumed that the experimental conditions, i.e. pressure, temperature and air motion have been established and the user-defined controlling variables entered into the computer. A typical screen display at the moment of fuel injection is presented in figure 2. Having reached this point the high-speed camera may now be started and this becomes the initiator for the rest of the system.

The sequence proceeds as follows, (figure 3). The high-speed camera accelerates to its maximum frame speed in about 0.5 sec at which point it outputs a signal to the computer via the data acquisition card. The computer then waits for the next pumping cycle reference signal. On receipt of this, the control and data acquisition sequence begins, this may be considered to be 'time zero'. The user will have defined two delay times (figure 2), the solenoid delay time and the delay before readings start. The former refers to the delay from time zero until the fuel pump rack solenoid is energised and is a method of controlling the injection characteristic. The delay before readings start, prevents the data acquisition card from obtaining data before anything interesting occurs. At the expiry of the solenoid delay, the solenoid is energised

for a user-selected duration (figure 2) and fuel injection occurs. Similarly when the reading delay ends, the data acquisition cycle is activated. The number of channels, number of samples and sampling rate having been set by the user, (figure 2).

At the completion of data acquisition the data held in the computer for channel 1 is displayed on the oscilloscope, typical examples of data held on channels 1 and 2 are shown in figures 4 and 5 respectively. Figure 4 represents the injector needle movement and figure 5 the variation in fuel pressure during injection. In both cases the period between samples is 30 micro-seconds. The oscilloscope display only shows a portion of the total data, however, the system allows the user to scan each channel at will to examine all the results. If the data is satisfactory it is transferred to tape and transmitted from a second microcomputer (equipped with an RS232 communications device) to the University mainframe computer for complete analysis. The original data presented in figures 4 and 5 is shown in figure 6 as plotted by the main-frame computer. (The injection needle movement has no scale since the needle always lifts 0.3 mm to the same physical stop). A second test diagram is presented in figure 7, in this experiment the chamber pressure and temperature at the start of fuel injection were 25 bar and 600k respectively. In this figure the injection event of figure 6 is compressed by the time scale and the chamber pressure (which is scaled by a factor of 10) indicates the onset of combustion by the rapid pressure rise after an initial preparation delay following fuel injection.

From these limited examples it is hoped to show the detail made available by the system and indicate its flexibility. In the following sections the major hardware components and the system software will be described.

SYSTEM HARDWARE

The Solenoid Timing and Control Card

The solenoid control card is shown schematically in figure 8 and consists of a 6522 VIA (Versatile Interface Adapter) and a 6840 PTM (Programmable Timer Module) with three 16 bit timers. The VIA (VIA-B) timers 1 and 2 (t1/2-B) are used to set the delay in initiating the analogue to digital converter (A/D). The required delay, entered into the computer, is set on timer 2 and on receiving the start signal, (time zero), timer 1 pulses at 1MHz counting down timer 2. When timer 2 is counted out the A/D card is activated.

The control of the solenoid delay and energise times is achieved by the PTM. On receiving the signal to start (time zero) on gate G1, countdown timer 1 outputs 1 MHz pulses from output A. Counter 2 (cdt2) has been armed with the solenoid

delay time and receives these pulses thus counting out the delay. On counting out, cdt2 opens the PULSE CONTROL gate allowing a 1 MHz signal into cdt3 which has been armed with the solenoid energise time. The solenoid is then energised until cdt3 counts out.

The Data Acquisition Card

The data acquisition card is shown schematically in figure 9 and consists of an Analogic MP6812 A/D and a 6522 VIA. The VIA (VIA-A) is used to control the A/D in terms of sampling rate, number of samples and number of channels. The VIA timers are used since to use the computer microprocessor (6502) would be too slow. Timer T1A is loaded with the sample interval (minimum 14 microseconds) and T2A with the total number of samples on all channels. The number of channels selected is also loaded into the VIA. The 6812 A/D has been set for short cycle operation in an 8 bit unipolar mode. The 6812 automatically cycles through the channels selected, upon receiving a strobe from timer T1A of the VIA. The channel currently sampled being represented in binary at the A/D output.

The method of operation is as follows. With the sample interval, number of channels and total number of samples loaded into the locations described above the system is ready. The camera speed signal enters the VIA control line CA2. Before the system can be fully activated, however, it requires another signal from the fuel pump. On receipt of this reference signal on CB1 of the VIA, the VIA starts timer T1A counting down at 1MHz until the sample interval (e.g. 15 microsecs) is counted out. On counting out, timer 4 T1A strobes the A/D initiating the start of conversion procedure. The pulse from T1A also decrements T2A (the total number of samples required) by one. The 6812 samples sequentially from the initial channel. At the end of conversion the digital data is presented to the VIA and the end of conversion flag (EOC) is set. On sensing the EOC the computer collects the data from the VIA (automatically clearing the EOC flag) and stores the data in memory. The computer increments its memory index and returns to wait for the next EOC. On receiving subsequent strobes the A/D multiplexer increments to the next channel until a coincidence occurs between the number of channels selected (held on the VIA) and the current channel sampled by the A/D. On this coincidence the reset line is forced low resetting the A/D converter to the initial channel and sampling continues until timer T2A times out. This interrupts the computer which then enters the oscilloscope output mode.

The Oscilloscope Output Card

This card is shown schematically in figure 10 and consists of 3 main circuits controlling the 'X' and 'Y' co-ordinates of the data output and the 'Z'-mod bright-up pulse. On

instruction from the computer the cards VIA (VIA-C) output reg B presents the digital representation of the 'X' co-ordinate to a digital to analogue converter (DAC). In a similar way output reg A supplies the 'Y' co-ordinate, simultaneously the 'Z' mod pulse is sent automatically by the VIA. The pulse is delayed to give the amplifiers time to settle and lengthened to provide adequate brightness. See figures 4 and 5.

SYSTEM SOFTWARE

Program Initialisation

The program is written in BASIC and MACHINE CODE and the combined flowchart is presented in figure 11. The two sections are concatenated together in memory so that the machine code is invisible to the user and in such a way as to allow the same program to be used in any ram size machine. BASIC is only used for inputting user parameters and saving the acquired data to tape. Machine code is used for the main data handling program due to its increased speed compared to BASIC. When the program is run the user is asked whether calibration or normal run mode is required (a calibration routine is also described later). If the normal run is selected the VDU display changes to its 'run' state (see figure 2). As figure 2 shows, the required input parameters are displayed in tabular form. Initially each parameter displays a default value, pre-set in the software, which may be accepted by the user by pressing the return key or altered using the backspace/rubout key. The user-accepted parameter value in the data window is checked for under- or over-range and any errors are detected and indicated by the computer, which then awaits a corrected input. All the data input is treated in this way, however, at the end of input an approximate run time is calculated and displayed. If this is not acceptable the user may return to alter any of the final three parameters. Once an acceptable run time is displayed and accepted the computer will ask the user to start the high-speed camera. The user run parameters are transferred to the machine code program which assumes complete control of the system. The computer sets up all the timer/counters described in the hard-ware section, calculates how much RAM is required to store the acquired data and reserves that amount at the top of RAM. A schematic representation of the resulting memory allocation is shown in figure 12.

Data Acquisition Mode

At this stage the program is sitting in a loop awaiting the signal indicating that the camera has reached maximum speed. When this occurs another loop is entered and the program waits for the next injection pump reference signal (see timing diagram, figure 3). When this is detected the solenoid energise delay and data read delay begin simultaneously. On

completion of the data read delay the data acquisition process is started. The program then waits in a loop for an end of conversion signal. On receipt of this, the data is stored sequentially in RAM and the store addressing pointer is incremented to the next RAM location. The program then returns to await the next end of conversion. This process is halted only by an interrupt from timer 2 on the A/D card (see section on hardware) which signals the end of data sampling.

Data Output to the Oscilloscope

As the amount of data acquired may be very large it is output to the oscilloscope for immediate examination. Data channel 1 is displayed by default with the other channels user-selectable. A 'window' of 256 data points width is set up, see figure 13, with the program effectively looping and continuously outputting this selected data. On selecting a channel the 'window' is set at the start of the data stream but may be moved along the data using the '<' and '>' keys (for the two directions of scan). Continuous depression of a key results in a continuous movement of the 'window' which freezes when the key is released. Output to the oscilloscope is through the oscilloscope output card described in the hardware section.

Having examined the experimental results, the user may either save the data to magnetic tape or reset the system to run from the program initialisation status described above.

Saving Data to Tape

A data file header is created containing details of the experiment followed by the experimental results. As speed is unimportant, data is normally saved to cassette tape at a rate of typically 600 baud. The complete data file now on tape is then transferred to the mainframe computer for further analysis via the second modified microcomputer.

Calibration Mode

The normal operating mode of the system is of a 'single shot' nature and in order to calibrate the system a special subroutine is included. This machine code routine scans all of the channels from the A/D and displays the channel values continuously, enabling calibration of the hardware cards and a check on the instrumentation amplifier outputs.

REFERENCES

Packer, J.P. (1983) Ph.D. Thesis, School of Mechanical Engineering, The University of Bath, England.

Packer, J.P., Wallace, F.J., Adler, D. and Karimi, E.R.(1983) Diesel Fuel Jet Mixing under High Swirl Conditions, Int. Conference on Combustion in Engineering, Proc.I.Mech.E.

ACKNOWLEDGEMENT

The work reported in this paper was supported by grants from the Science and Engineering Research Council.

FIGURES

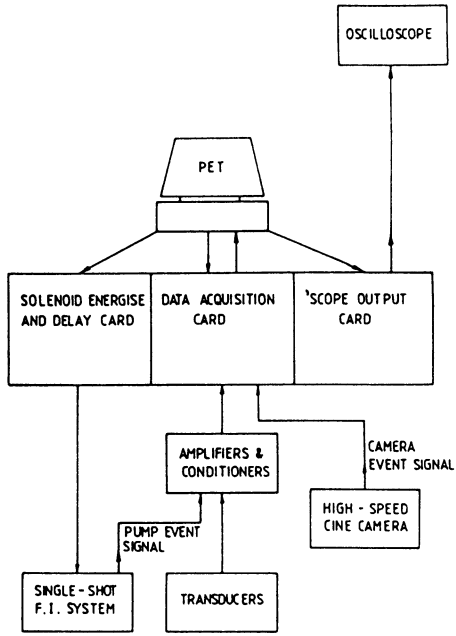


FIGURE 1 RIG CONTROL AND DATA ACQUISITION SYSTEM

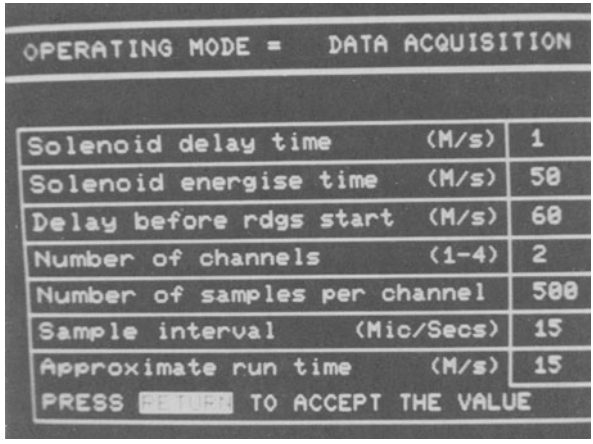


FIGURE 2 COMPUTER DISPLAY

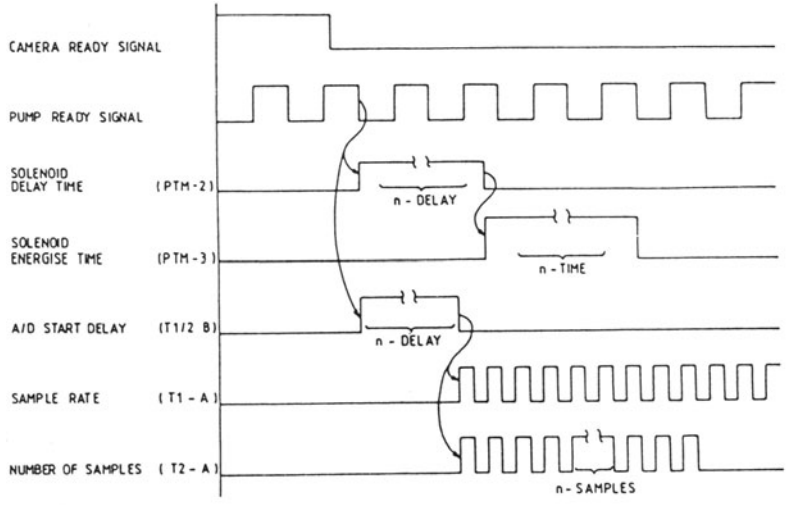


FIGURE 3 SYSTEM TIMING DIAGRAM

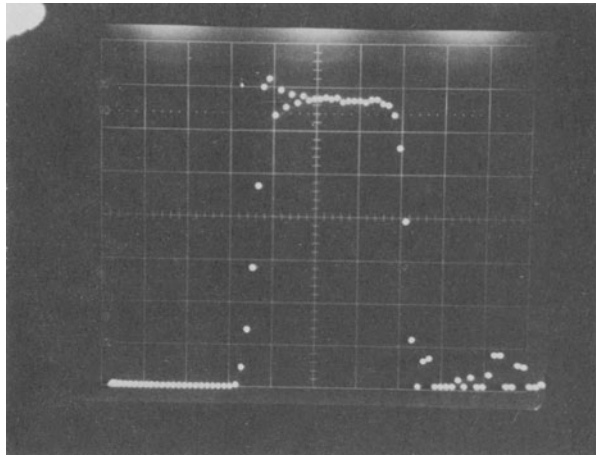


FIGURE 4 OSCILLOSCOPE DISPLAY
CHANNEL 1 - NEEDLE LIFT

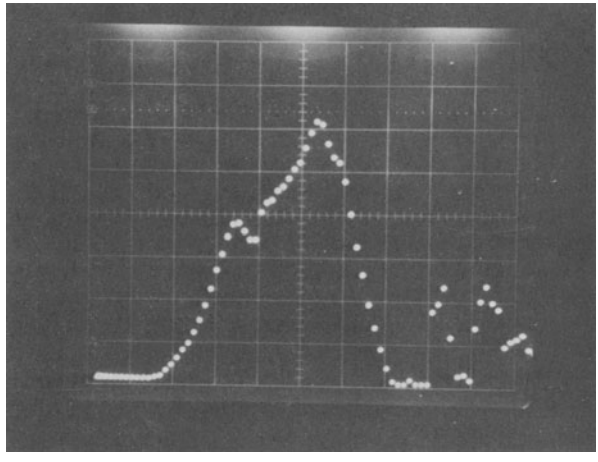


FIGURE 5 OSCILLOSCOPE DISPLAY
CHANNEL 2 - FUEL PRESSURE

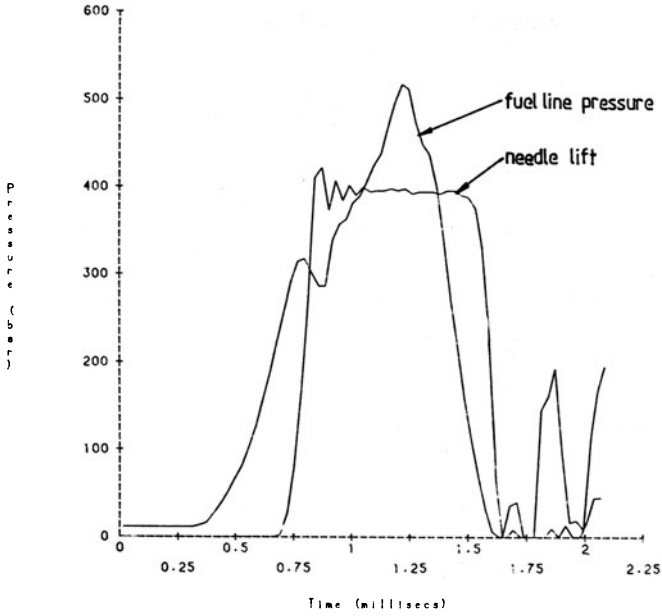


FIGURE 6 FUEL INJECTION CHARACTERISTIC

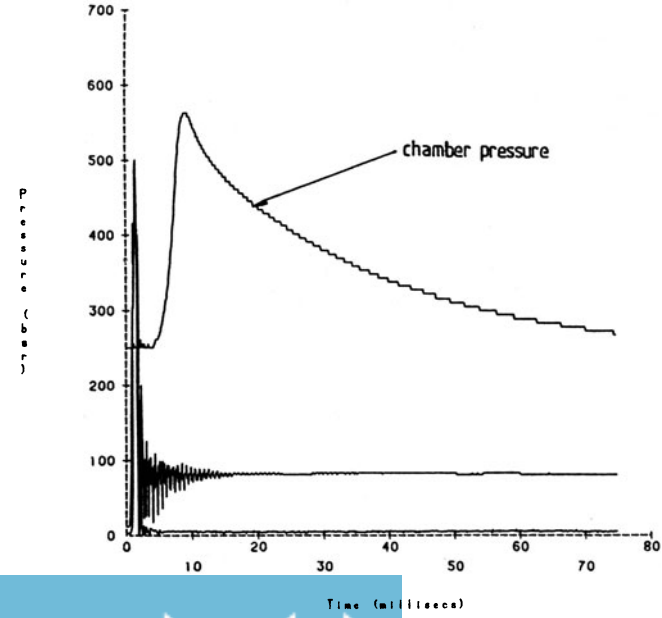


FIGURE 7 COMBUSTION TEST

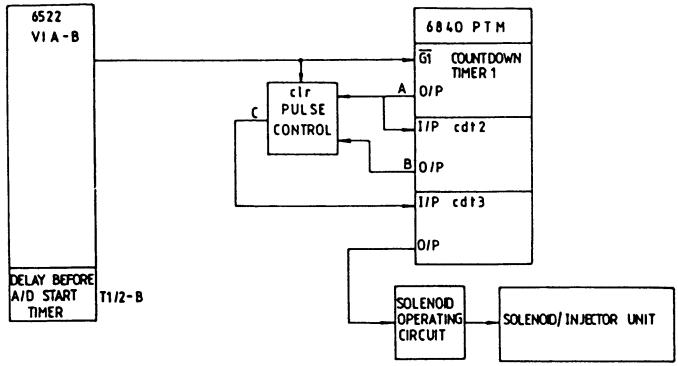


FIGURE 8 SOLENOID CONTROL CARD

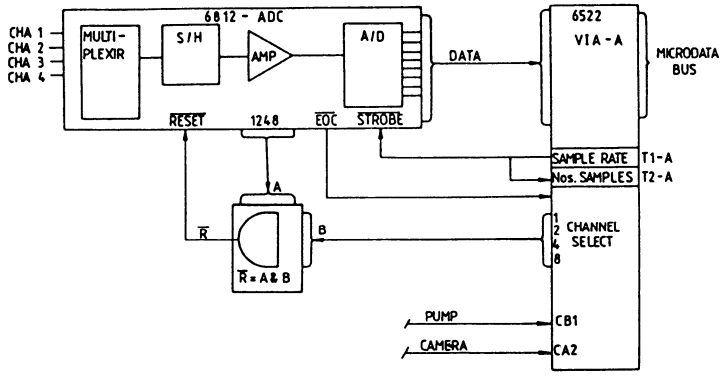


FIGURE 9 DATA ACQUISITION CARD

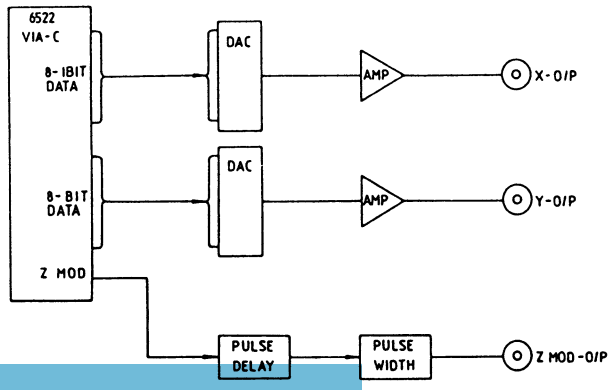


FIGURE 10 OSCILLOSCOPE OUTPUT CARD

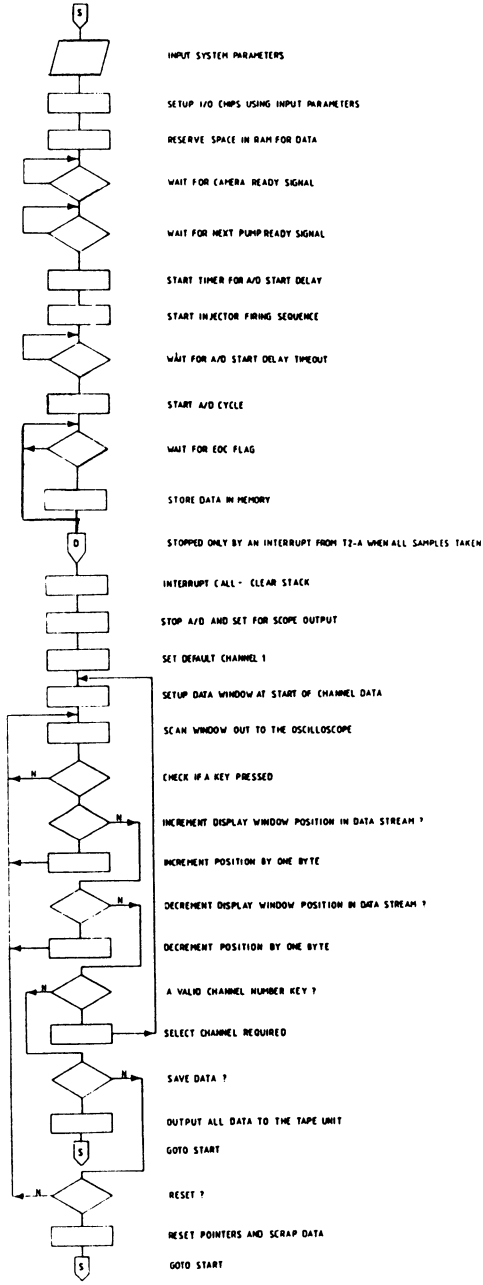


FIGURE 11 PROGRAM FLOW-CHART

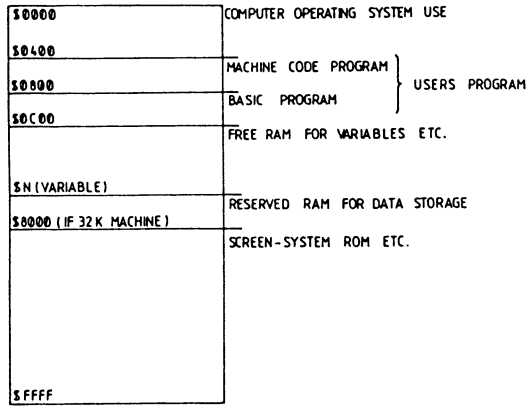


FIGURE 12 COMPUTER MEMORY ALLOCATION

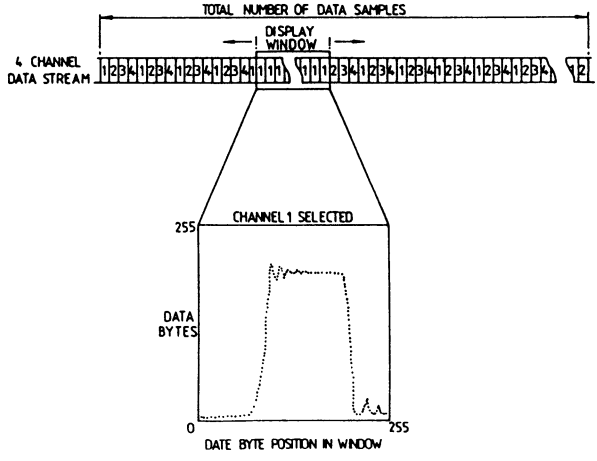


FIGURE 13 DATA OUTPUT WINDOW

P.A.I.D. - AN INTERACTIVE GRAPHIC PACKAGE TO SUPPORT PIPING ANALYSIS

Zs. Révész

Numerical Analysis Section
Electrowatt Engineering Services, Ltd.
Zurich, Switzerland

ABSTRACT

PAID is an acronym for Piping Analysis and Interactive Design. It is an outgrow of a number of small programs written to facilitate piping structural engineering with the commercially available code PIPESTRESS.

The installed and tested capabilities of PAID are presented in this paper. Examples are given from the graphic output in form ready to be included in stress analysis reports. Sample conversations necessary to produce such plots are given also. The attached graphic output covers:

- Structure layout representation with isometric and plane view including support system plot.
- Graphic representation of structure loads both, of response spectrum type and of time history type.
- Representation of results of spectral analysis.
- Graphic postprocessing for documentation of static displacements, mode shapes, bound solution displacements of dynamic load cases, stresses, reaction forces and moments at the supports.

Other capabilities of the program are briefly described. These include interactive input of structure and loads, generation of various data sets and the manipulation of such.

The paper summarizes the experience gained on the way towards interactive piping structural engineering with full graphic output.

INTRODUCTION

The kind of projects managed by Electrowatt Engineering Ser-

vices Ltd., involves integral design of piping systems with supports and the qualification of such for various postulated events. The large number of systems, the growing complexity of the load cases involving seismic/fluid-dynamic excitation, explosions and aircraft crash require human assimilation of large amounts of data. An effort has been made to enlighten evaluation of numerical data and visualize as much of it as possible, thus eliminate a source of error and accelerate analysis/reporting. The product of this effort is PAID.

These special circumstances of the development (i.e. being strongly application oriented and having no extra budget) influenced the results determinatively. A program has emerged which is well tested, is efficient and tailored for the needs of piping/support structural analysts. The specific features of the package are:

- Past and present strategy of development (i.e. new capabilities will be built in only if needed by an analyst and such installation are usually completed by the test and/or production runs).
- Output form (i.e. preparing plots for direct inclusion into reports).
- Program logics and execution mode (that means interactive execution providing built-in help and error checks at nearly every stage of execution).
- Documentation (which is mainly built into the program, limiting the program manual to the description of capabilities and how to start working with the code).

It has not been tried to make the analysis itself by PAID. Interest has been focused on speeding up work done mainly by PIPESTRESS. This goal has been achieved by installing a wide variety of graphic output options.

HISTORY OF DEVELOPMENT

PAID is an acronym derived from Piping Analysis and Interactive Design. It has been developed at Electrowatt Engineering Services to facilitate piping structural engineering work performed for various power plants. The analysis program package which is extensively used by the company is EBASCO/PIPESTRESS. It is known as one of the best suited software products for the work of this kind. Its theory is well documented, see e.g. Gordis (1979, 1981a), it is widely applied and is suitable for economic analysis of special problems as reported by Gordis (1981b) and Révész (1982, 1983).

The fact that there are numerous users of PIPESTRESS, promoted its development and increased its reliability. But this wide usage made tedious the installation of specific requests from companies with respect to data processing.

Having realised this, work was started at Electrowatt to facilitate the use of PIPESTRESS with in-house terminal hardware/software. The acronym "paid" has been chosen to refer to the circumstances of the development i.e. its development has been an investment of our company in order to improve performance. Though the program is commercially exploitable, the primary aim was never to build a program for marketing purposes. It has rather been attempted to solve specific problems of individual analysts in the frame of a system. This ensured continuation, so that solution methods have been documented and reserved for quick solution of the problem on a subsequent occasion. In this light the PAID software is an outgrow of a number of small programs and is still in continuous development corresponding to the needs arising from Electrowatt's activities on the field of structural analysis.

Growth of the software occurred in a random, spontaneous manner. This unplanned growth implied no restraints (except funding) and resulted in an efficient development procedure. The PAID software, a by-product of Electrowatt's activities, has become in this way secured and valuable know-how of the company.

Though developed for our own purposes, utilization and development were kept separated. This enabled an efficient error testing through criticism and economic installation of new features through feedback. In each case the development request, if realistic, has been fulfilled by installation of the new option with subsequent testing by the originator. Since analysis work is increasingly performed by engineers rather then by computer specialists, user-friendly execution with a clear question-answer program flow has been emphasized. Much assistance is the user provided with, when carrying out piping analysis. This assistance consists of internal checks, error messages and explanations if an answer were illegible for the program. Consequently, the program documentation is mostly on-line. Instructions are given and questions are asked, during interactive execution. The hardcopy documentation is minimal and is basically limited to the correct call sequence, sample results and sample conversations.

The outlined background of the program development influenced the results determinatively. The particular features of the software product can be summarized as follows:

- PAID is well tested.
- The strategy of development has been straightforward, dictated by the needs.
- The output is in report form, plots are properly spaced, texts are legible, sheets dated and identified.
- Interactive execution through conversation with the program in plain English language.

- Simple logics with numerous internal checks during execution accomodating users who are less experienced in the use of computers.
- Minimal written documentation with maximum comfort and user support during execution.
- Use of procedure file for repetition of actions and for batch execution.
- Quick engineering interpretation and understanding of the elaborated results through manifold graphic capabilities.

PRESENTATION OF RESULTS

The danger of inappropriate use of a structural analysis program can be lessened by extensive use of computer graphics. Engineering misapplications cannot be prevented but their potential is greatly reduced by the use of graphic interpretation. This optical control is possible and necessary at all stages of a project, from the beginning of general planning, through mesh generation, load input, result representation until the preparation of adequate documentation. At all these stages a flexible and interactive graphic system can be of immense help. Such a system permits the user to examine many details of the model before a solution is effected, and then to interpret the results after the problem has been solved.

It is seldom that a larger system can be analysed in a single run. For example spectral analysis or static analysis may lead to the change of the system before proceeding to the next step. The interactive graphic capabilities of PAID in connection with the restarting capabilities of PIPESTRESS enable a rational procedure and economic analysis of advanced piping problems.

Four major segments of PAID are released by now, dealing with structure layout, loads, spectral analysis and results. To each segment one or more plots are given and explained below.

Layout

Using PAID, a piping structure can be input into the computer storage in different ways. Once stored, it can be modified as per necessity. Further, the program enables checks of a certain layout defined by an input deck. The checks include listing of a point table and generating different plots of isometry. The installed options enable the analyst both, to have an overall view of the entire piping system as a single entity and to plot specified segments of any size for documentation. The point of view can be chosen arbitrarily, the plots are proportionally scaled and (as described in connection with the piping analysis results later in this section,) computed results such as forces and displacements can be blended with the structure layout. This enables the analyst to depict realistic geometrical configurations of the

```
\paid,cards
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? layout
WHAT DO YOU WISH TO DO WITH THE LAYOUT
? plot
 TO PLOT WHAT
? supports
 A FULL ISOMETRY
? yes
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? no
DO YOU WANT TO REDEFINE THE AXES
? no
 BYE
 JOB REPRIEVED.
```

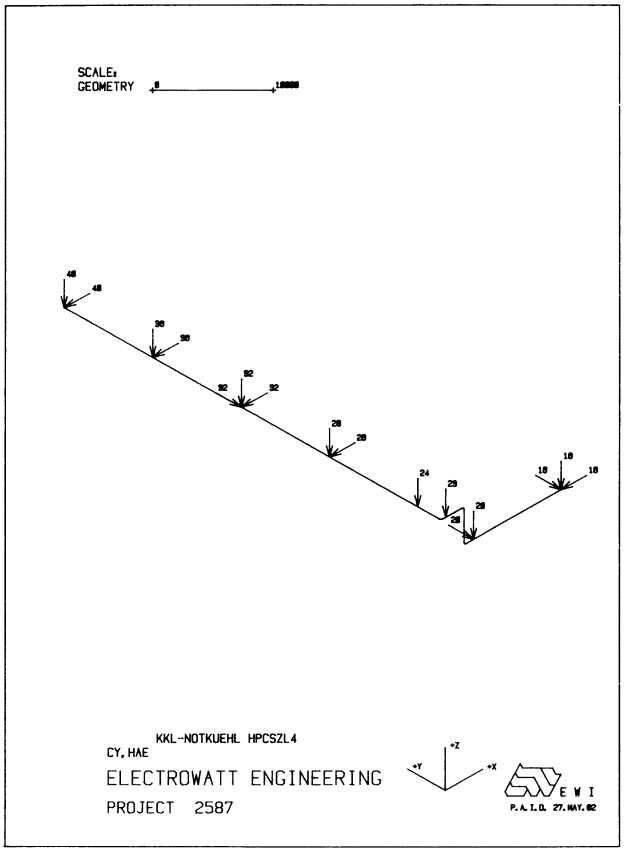
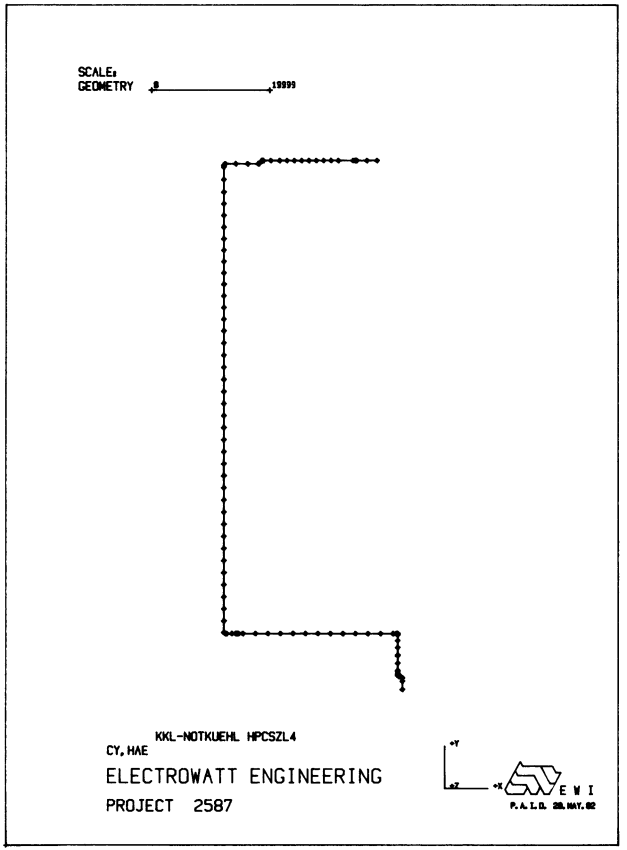


Figure 1. Isometric view of pipe routing with restraints

```
\paid,cards
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? layout
WHAT DO YOU WISH TO DO WITH THE LAYOUT
? plot
 TO PLOT WHAT
? isometry
 A FULL ISOMETRY
? yes
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? yes
DO YOU WANT TO REDEFINE THE AXES
? yes
ANGLE OF "X" TO HORIZONTAL
(360. SUPPRESSES THIS AXIS)
? 0.
ANGLE OF "Y" TO HORIZONTAL
(360. SUPPRESSES THIS AXIS)
? 90.
ANGLE OF "Z" TO HORIZONTAL
(360. SUPPRESSES THIS AXIS)
? 360.
 BYE
 JOB REPRIEVED.
```

Figure 2. Plane view of a system indicating the nodes



system and to place restraints at structurally convenient locations. There are two sample plots attached with corresponding conversation. Figure 1 shows pipe routing with translational restraints, and Figure 2 a plane view of a piping model indicating the nodes.

Loads

The loads acting on a piping have been divided into response spectrum type of loading and time history type of loading. This division has been made strictly on the basis of the analysis method used.

The response spectra are of crucial importance in case of dynamic analysis. PAID is interfaced with PIPESTRESS in order to provide opportunity for input, graphic representation, controlling, combining and preparation of input data. The features implemented so far include:

- Plotting of all response spectra stored in a PIPESTRESS input deck (with pre-defined scaling if necessary).
- Combination of floor response spectra to get the response spectrum at a given restraint which is located at a certain distance from the lumped mass of the floor.
- Broadening of a spectrum.
- Manual and analog input of spectra.

Figure 3 gives an example from the graphic output of an input deck.

Analysis of dynamic effects in the time domain requires definition and storage of force time-histories at the active nodes. Amoung the data structures used for storage of time-dependent data is the Time History Applied Force File (THAFF) of PIPESTRESS. The input options are general alphanumeric cards, PIPESTRESS-compatible card format and binary THAFF format. Output options are different card decks, listings, plots and binary THAFF format. The capabilities installed so far are creation, listing, modification, plotting, combination of different time-histories and the conversion of pressure time-histories into forces. Figure 4 depicts the force input at a node.

Spectral analysis

The response of an elastic system r(t) is different from the applied load f(t). The total response can be obtained by summing up all differential responses developed during the load history. It is very convenient to define the dynamic loading factor as

$$DLF = \max |r(t)| \ / \max |f(t)|$$

and the dynamic part of loading as

$$DPL = \max |r(t) - f(t)| \ / \max |f(t)|$$

where DLF and DPL are both depending on the resonance frequency of the system in question. The plots of these functions, called spectra, are suitable for assessment of fre-

```
\paid,cards
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? loads
 ARE YOUR LOADS OF RESPONSE SPECTRUM TYPE
? yes
 WHAT DO YOU WANT TO DO WITH SPECTRA
? plot
DO YOU WANT TO USE THE SCALE OF THE FIRST PLOT
 FOR ALL SUBSEQUENT ONES
? no
BYE
 JOB REPRIEVED.
```

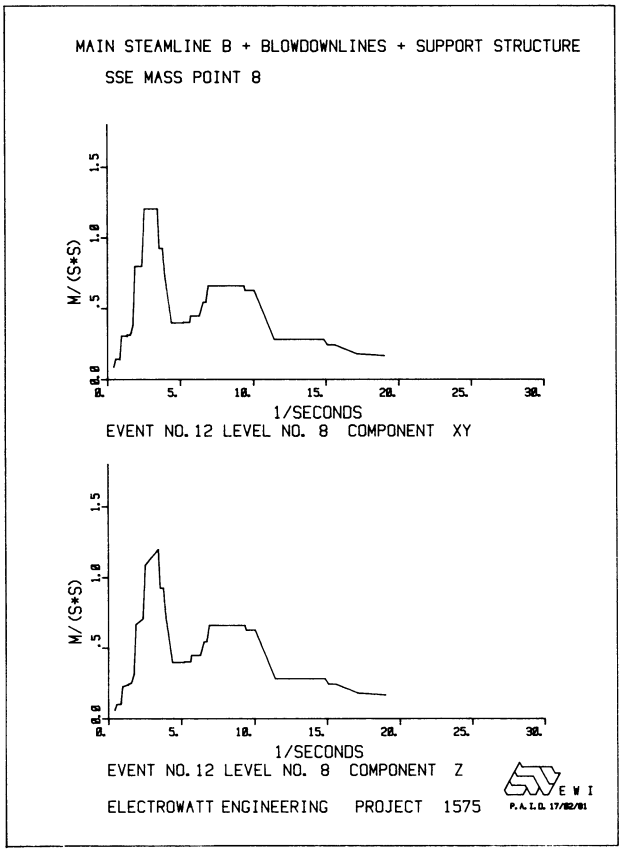


Figure 3. Earthquake response spectra from an input card deck

```
\paid,,,thaff
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? loads
 ARE YOUR LOADS OF RESPONSE SPECTRUM TYPE
? no
 IS THE PIPESTRESS VERSION YOU USE 3.2
? yes
WHAT DO YOU WANT TO DO WITH YOUR THAFF ?
? plot
ARE YOUR DATA STORED FOR THE INDIVIDUAL
   SEGMENTS AS IT HAS BEEN SPECIFIED IN
   PAID MANUAL SEC. 3.21
? no
ARE YOUR DATA STORED FOR COMPONENTS
ON A THAFF AS FOR DST/PIPESTRESS
? yes
WHAT IS THE NO. OF PLOTS ON ONE PAGE HORIZONTALLY
? 1
WHAT IS THE NO. OF PLOTS ON ONE PAGE VERTICALLY
? 1
DO YOU WANT TO GET ALL THESE ON A SINGLE PAGE
? yes
 BYE
 JOB REPRIEVED
\
```

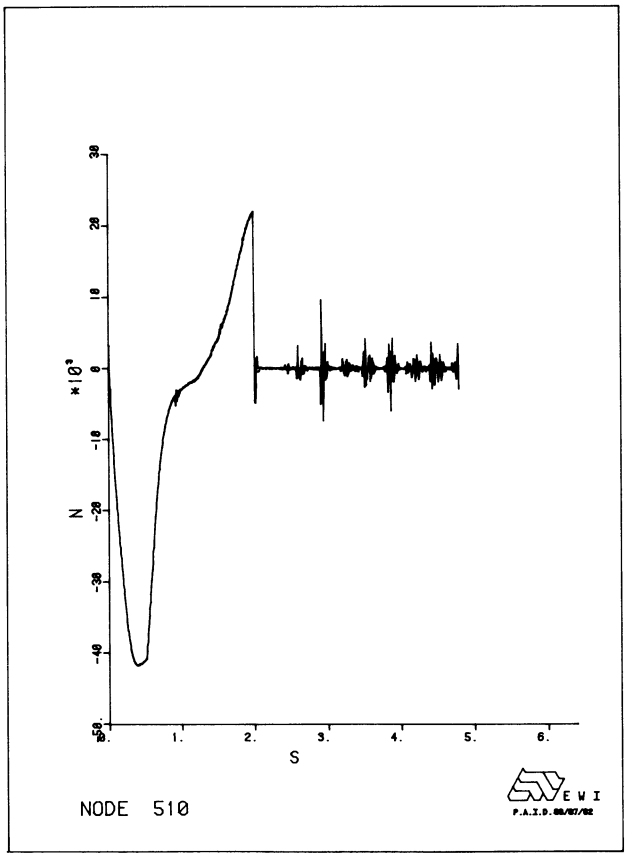


Figure 4. Force time-history at a node

```
\paid,rxout
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? spectra
WHAT IS THE ORIGIN OF YOUR SPECTRA
? hm...
YOU ARE EXPECTED TO ANSWER
 "SPECT" OR "PSRX"
WHAT IS THE ORIGIN OF YOUR SPECTRA
? psrx
  BYE
  JOB REPRIEVED
/
```

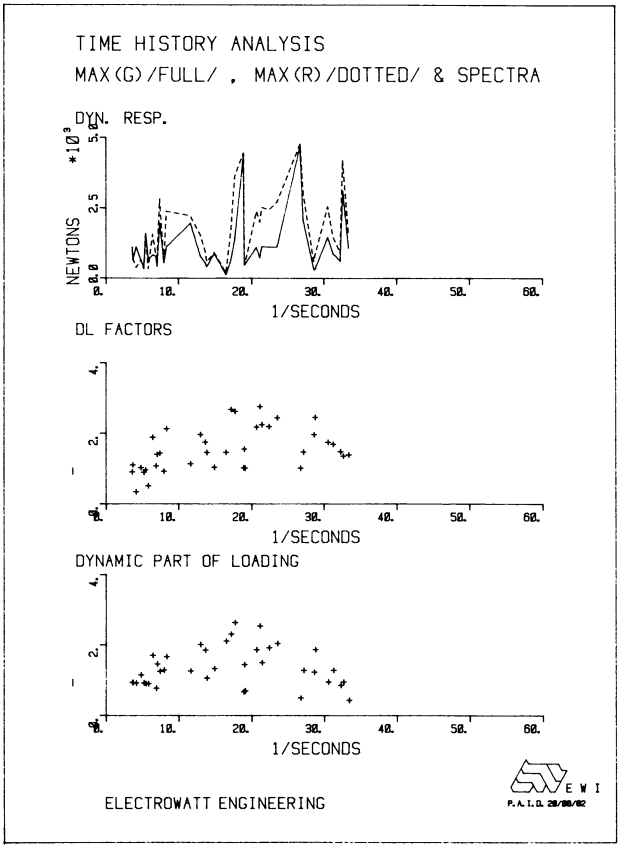


Figure 5. Maximum generating force (G), maximum response (R) and their relationship

quency content of a certain load.

There is an option of PAID to plot these spectra. The purpose of this optical evaluation can be either investigation on different load time-histories independently from a structure using single degree of freedom models, or checking if a solution obtained by modal superposition in a certain frequency range is reasonable if the loads are interacting with the structure. This second kind of spectra is computed using the complete finite element model of the system represented at the individual resonance frequencies by the corresponding mode shapes, and the applied force is the complete force time-history vector. The background of such a generalized response spectrum is explained in more detail e.g. by Clough and Penzien (1975), Gordis (1981b) and Révész (1982). A sample plot is given in Figure 5. On the upper plot of Figure 5 the maximum generating force G is shown as seen by the structure at the different resonance frequencies. On the same plot the response R is indicated too, again as seen by the system. The generalised DLF-s and DPL-s i.e. spectra are given at the centre and the bottom.

Results

To enlighten the work of the analyst and to reduce the work connected with documentation PAID has been prepared to visualize as much from the tabulated output of PIPESTRESS as possible.

Options available are
- plot of displacements in static load cases,
- plot of mode shapes,
- plot of bound solutions for dynamic load cases, and
- plot of reaction forces in individual load cases.

The attached plots on Figures 6 through 9 advocate themselves. They can be requested one-by-one in an interactive manner as soon as the analysis program has performed the computation, or they can be requested in a batch job connected with the analysis run.

SUMMARY OF EXPERIENCES

The use of a program system like PAID enables offering of competitive services. Economically it offers very good prospects because the hardware prices are continuously decreasing and because it enables a radical reduction of the hardcopy documentation i.e. no large outputs will be stored and distributed for control any more. Other advantages are the very fast result representation, the ease of evaluation of large systems and the support of analysts at solving detail problems.

```
\paid,haeinp,,,,,haeout
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? results
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? no
DO YOU WANT TO REDEFINE THE AXES
? no
WHICH DATA
? mode shapes
 IS THE FACTOR OF DEFORMATION 1.
? yes
WHAT IS THE MODE NO.
? 2
WHICH DATA
? none
 JOB REPRIEVED.

\
```

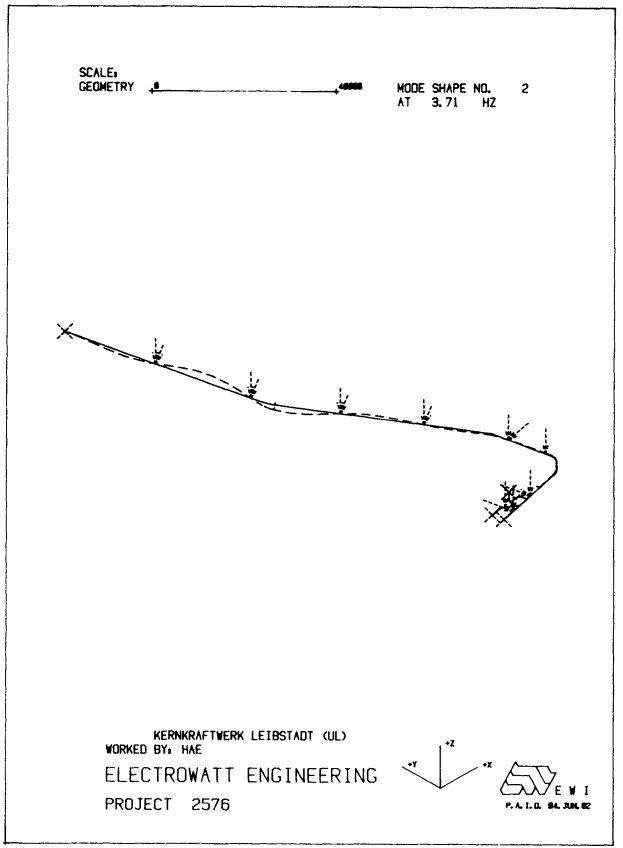


Figure 6. A typical mode shape

```
/paid,haeinp,,,,,haeout
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? results
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? no
DO YOU WANT TO REDEFINE THE AXES
? no
WHICH DATA
? displacements
 WHAT IS THE LOAD CASE NUMBER
? 1
TYPE OF LOAD CASE NO.    1
? lc
WHAT IS THE MAGNIFYING
FACTOR FOR DISPLACEMENT
? 200.
WHICH DATA
? none
 JOB REPRIEVED.

\
```

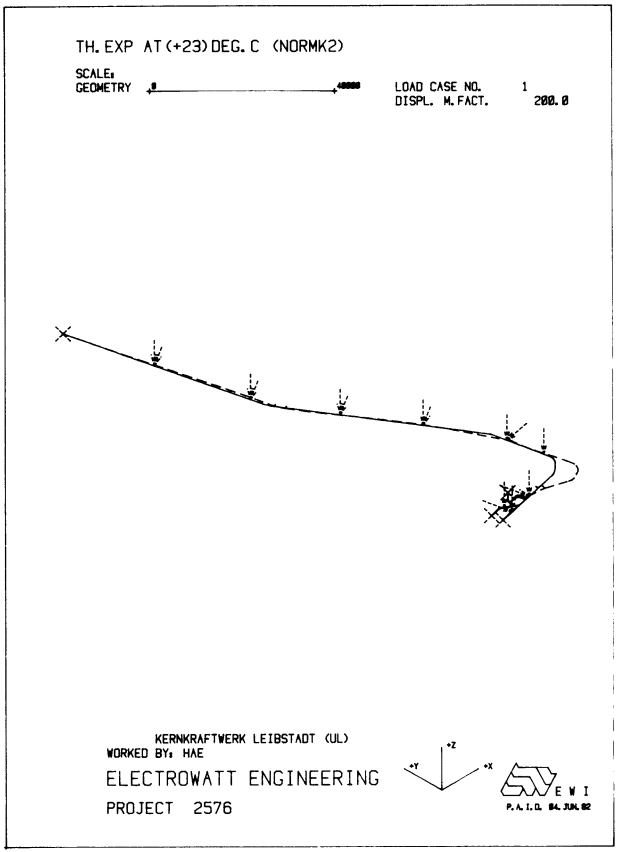


Figure 7. Graphic representation of a static load case

```
\paid,haeinp,,,,,haeout
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? results
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? no
DO YOU WANT TO REDEFINE THE AXES
? no
WHICH DATA
? displacements
 WHAT IS THE LOAD CASE NUMBER
? 11
TYPE OF LOAD CASE NO.  11
? rc
WHICH DATA
? none
 JOB REPRIEVED.
/
```

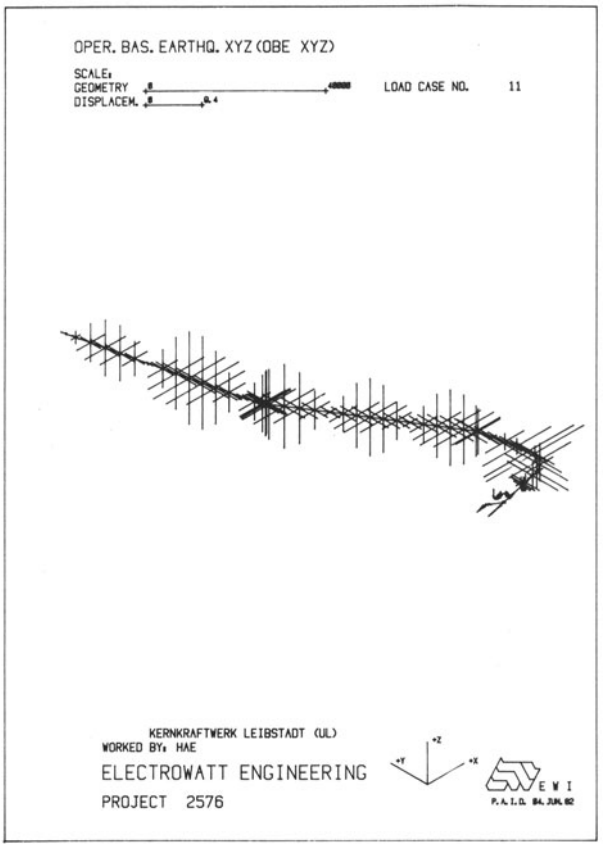


Figure 8. Solution of a dynamic load case

```
\paid,haeinp,,,,,haeout
HELLO PAID-USER
WHAT'S THE SUBJECT OF YOUR ACTIVITY
? results
IS A4 THE SIZE YOU LIKED
? yes
DO YOU WANT TO WRITE ALL THE NODE NUMBERS
? no
DO YOU WANT TO INDICATE ALL THE NODES
? no
DO YOU WANT TO REDEFINE THE AXES
? no
WHICH DATA
? forces
LOAD CASE NUMBER
? 111
WHICH DATA
? none
 JOB REPRIEVED.
```

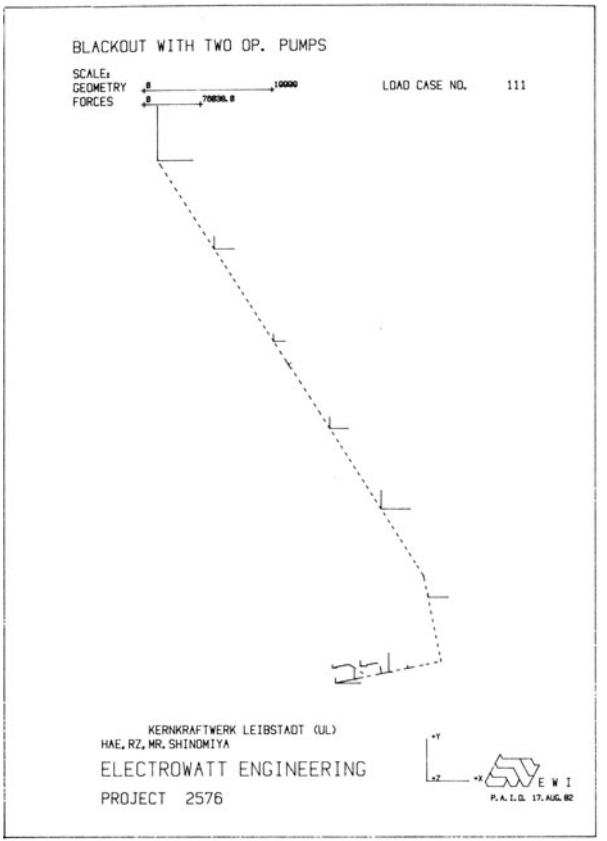


Figure 9. Plot of reaction forces

The result of our development, the user friendly PAID, has made finite element analysis more convenient and more feasible. In this way it helped to expand the group of program users from computer specialists to other members of the engineering community. Making advanced engineering software accessible to a larger fraction of engineers is not only dictated by necessity, but may also be decisive in the competition between companies.

The PAID software is sufficiently documented and is installed in a commercial computing centre with international network. There are negotiations about public release. At the time of manuscript preparation the program was available to users only on basis of bilateral agreements.

REFERENCES

Clough R.W., J. Penzien (1975) : Dynamics of Structures, Tokyo, McGraw-Hill Kogakusha

EBASCO/PIPESTRESS, Version 3.2, USER'S GUIDE Geneva, DST Computer Services S.A., 1981

Gordis K. (1979) : Outline of Dynamic Analysis for Piping Systems, Nuclear Engineering and Design, Vol. 52, No. 1, pp. 99-110

Gordis K. (1981a) : Mathematical Basis of Dynamic Structural Analysis Methods for Piping Systems, Denver, ASME Joint Conference Proceedings

Gordis K. (1981b) : Generalized Response Analysis : A Method for Calculating Bound Solutions for Elastic Structures Subject to Dynamic Applied Force Loads (Trans. of 6th Int. Conf. on SMIRT), Amsterdam, North-Holland

PIPING ANALYSIS AND INTERACTIVE DESIGN, Version A/2, USER'S MANUAL, Zurich, Electrowatt Engineering Services Ltd., 1982

Révész Zs. (1982) : Analysis of Relief/Safety Discharge Loads by Modal Approach, Orlando, International Modal Analysis Conference

Révész Zs. (1983) : Comprehensive Stress Analysis of BWR Discharge Lines Subjected to High Frequency Fluid Dynamic Excitation (under press) Int. J. for Pressure Vessels and Piping

RESOLUTION OF THE STATIC AND DYNAMIC PROBLEM IN ELASTIC CABLES THROUGH A VARIATIONAL PRINCIPLE

López, A. (*); Tárrago, J.A. (**); Bastero, C (**); Bastero, J.M. (*).

(*) E.T.S. Ingenieros Industriales de San Sebastián
(**) E.T.S. Ingenieros Industriales de Bilbao

This paper introduces a functional based on the generalized principle, which is useful for the static and dynamic study of cables and elastic-linear cables. The general nature of the loads avoids the usual simplifying hypothesis.

The finite elements method (F.E.M.) is used to obtain numerical results. The three-node parabolic element is used in this study, making it possible to obtain satisfactory results. Nevertheless, depending on the type of problem used and on the loading system, other kinds of elements may be used. The initial configuration necessary for the iterative process of calculating the static equilibrium position is automatically selected, thus avoiding previous calculations.

In order to solve the static problem, taking into account the non-linear nature at the equations, we used the Newton-Raphson method, the convergence of which has been proved in every case. In the dynamical problem, the eigenvalues and eigenvectors were obtained by means of a combined method of inverse iteration-bisection. In the inverse iteration, the Gram-Schmidt ortogonalization method was used to separate the multiple frequencies.

INTRODUCTION

The study of cables has long been one to stimulate researchers' interest. Prominent figures from the world of mechanics during past centuries - people such as Galileo, I.Bernouilli, Euler etc ... studied different problems related to cable dynamics although the results they obtained in many cases were of no more than a pure academic interest.

The complexity of the differential equations when considering the possible lengthening of cables showed that it would be better to abandon the attempts to find exact analytical solutions, an approach which had so attracted mechanics in the past (Routh, 1891; Pugsley, 1957). Research was now redirected to the establishment of different approximation techniques, in all of which the cable - which is a unidimensional continuum and, as such, exhibits infinite degrees of freedom - is replaced by a discrete mechnaical system which is presupposed to express the behaviour of the cable with sufficient concordance (Wilson, 1977; Michalos y Birnotie 1962; O'Brian y Francis, 1964; Peyrot y Goulois, 1979).

In this way, the P.E.M. has given way to exceptional possibilities, not only because it is based on the same variational principle that leads us to the exact analytical solution but also because, by choosing the interpolation functions properly, the solution obtained converges when the discretization is refined. It is, therefore, possible to study the critical zones more precisely or to limit the margin of error.

In this paper, which is restricted to continuous unidimensional elastic-linear media (which are, therefore lacking in stiffness on flexion), we shall start from the variational principle which establishes that the movement of a system of this nature confers a stationary value to the functional (Bastero, 1980).

$$J=\int_{t_1}^{t_2}\int_{0}^{l_o}\left\{\frac{1}{2}\,l_o\,A_o\{x\}^T\{x\}+l_o\,A_o\{x\}^T\{f\}+\{x\}^T|w|\{x'\}+\right.$$

$$\left.+\{x\}^T\{F_\alpha\}\bar{\delta}(S_o-S_\alpha)-\frac{1}{2}\,E\,A_o\left[1-(\{x'\}^T\{x'\})^{1/2}\right]^2\right\}dS_o\,dt \qquad (1)$$

where

I_o = natural length of the unstretched cable

A_o = transversal section of the unstretched cable (it is assumed that this is very small compared to I_o)

P_o = density of the unstretched cable

dS_o = length element of the unstretched cable

E = Young module

{x} = position vector of a generic point on the cable, at any moment, with respect to an absolute reference. It is evident that:

$$X_r = X_r\ (S_o, t) \qquad (2)$$

where S_o is the curvilinear coordinate (length of the unstretched cable from the initial point) at the point on the cable whose position is given by {x}; and t is time:

$$\{X\} = \frac{\partial \{X\}}{\partial t} \qquad \text{or} \qquad \dot{X}_r = \frac{\partial X_r}{\partial t} \qquad (3)$$

$$\{X\} = \frac{\partial \{X\}}{\partial S_o} \qquad \text{or} \qquad X'_r = \frac{\partial X_r}{\partial S_o} \qquad (4)$$

{f} = forces by unit of mass.

{w} = forces by unit of span.

$\{f_\alpha\}$= punctual force acting on a point of the cable characterized by the curvilinear coordinate S_α.

$\bar{\delta}(S_o - S_\alpha)$ = Dirac function.

STATIC PROBLEM

In order to study the equilibrium configuration of a cable under the action of a given force system, it is sufficient to require that the functional take on a stationary value:

$$W = \int_o^{l_o} \left\{ \rho_o A_o \{x\}^T \{f\} + \{x\}^T |w| \{x'\} + \{x\}^T \{F_\alpha\}\ \bar{\delta}(S_o - S_\alpha) - \frac{1}{2} E A_o \left[1 - (\{x'\}^T \{x'\})^{1/2}\right]^2 \right\} dS_o \qquad (5)$$

and the predetermined boundary conditions for the cable must be imposed on the solution which has been obtained in this manner.

The application of the F.E.M. requires defined discretization of the cable with n finite elements defined by the curvilinear coordinates of its ends: (S_i, S_i+i) for element i; and it defines various

interpolation functions $|N_i|$ for each element i, with which any point of the element τ can be specified in function of the nodal displacement vector:

$$\{x_i\ (S_o)\} = [N_i(S_o)]\ \{X_i\} \tag{6}$$

In this way the functional w can be replaced by the $\bar{w}$

$$\bar{W}= \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} \rho_o A_o \{X_i\}^T[N_i]^T\{f\}+\{X_i\}^T[N_i]^T[W][N_i]$$

$$\{X_i\} + \{X_i\}^T[N_i]^T\{f\} + \bar{\delta}\ (S_o-S_\alpha) - \frac{1}{2} E A_o \left[1 -\right.$$

$$\left.-\{X_i\}^T[N_i']\ \{X_i\})^{1/2}\right]^2 \Bigg\} dS_o \tag{7}$$

where

$$[N_i'] = \frac{\partial}{\partial S_o}\ [N_i(S_o)] \tag{8}$$

The stationary condition for $\bar{W}$ demands that:

$$\delta\bar{W} = o \tag{9}$$

and so:

$$O= \sum_{i=1}^{n} \{\delta X_i\}^T \int_{S_i}^{S_{i+1}} \Big\{\rho\ A_o[N_i]^T\{f\}+[N_i]^T[W][N_i']\{X_i\}+$$

$$+[N_i]^T\{F_\alpha\}\ \bar{\delta}(S_o-S_\alpha) - E\ A_o([N_i']^T\ [N_i] -$$

$$- \frac{[N_i']^T\ [N_i]}{(\{X_i\}^T\ [N_i][N_i']\{X_i\})^{1/2}}) \{X_i\}\ dS_o \tag{10}$$

This expression can also be written as follows:

$$O = \{\delta X\} \left(\{F_n\} + [K_w]\ \{X\} + \{F_p\} - [K_L]\ \{X\} - [K_T]\{X\}\right)$$

(11)

where

{X} holds for the nodal displacement vector.

$$\{F_m\} = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} \rho_o A_o [N_i]^T \{f\} \, dS \tag{12}$$

represents the nodal forces equivalent to the massic forces.

$$\{F_p\} = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} [N_i]^T \{F_\alpha\} \bar{\delta}(S_o - S_\alpha) \, dS \tag{13}$$

nodal forces equivalent to the exact forces. These operations can be avoided if the application points of these forces are chosen as nodes.

$$[K_w] = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} [N_i]^T [W] [N_i'] \, dS_o \tag{14}$$

$$[K_L] = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} E A_o [N_i']^T [N_i'] \, dS_o \tag{15}$$

$$[K_T] = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} - E A_o \frac{[N_i']^T [N_i']}{(\{X_i\}^T [N_i']^T [N_i'] \{X_i\})^{\frac{1}{2}}} \, dS_o \tag{16}$$

In order to solve the static problem it would be sufficient to impose the boundary conditions on (11) (generally, as predetermined nodal coordinates). The equilibrium equation is obtained in this way:

$$([K_L] + [K_T]) \{X\} = \{F_m\} + \{F_p\} + [K_w] \{X\} \tag{17}$$

From which the pre-established nodal coordinates in {X} have been eliminated - making the elements of this vector independent of one another - as well as the corresponding rows and columns in the remaining vectors and matrices.

The non-linear nature of this equation means that an iterative procedure is required in order to solve it. This procedure is explained more thoroughly in section 5 of this paper.

DYNAMIC PROBLEM, NATURAL FREQUENCIES

The most general movement of a one-dimensional elastic continuum can be solved in a similar way taking functional j as the initial point, Equation 1.

However, due to its fundamental interest, the dynamics study will be limited to the small oscillations of the cable around the equilibrium configuration. In this case the following change of variables will be made in Equation 1:

$$\{X\} = \{\bar{X}\} + \{u\} \tag{18}$$

where $\{\bar{X}\}$ holds for the non-stable configuration, thus making the functional in Equation 5 stationary, and {u} holds for the small displacements from that configuration. Obviously, as these small movements are caused by an infinitesimal disturbance of the system's mechanical energy:

$$\{u\} = \frac{\partial\{u\}}{\partial t} \tag{19}$$

and

$$\{u'\} = \frac{\partial\{u\}}{\partial S_o} \tag{20}$$

will also be very small values.

By substituting the varialbes from Equation 18 in Equation 1, the following functional is obtained:

$$J = \int_{t_1}^{t_2} \int_{o}^{l_o} \frac{1}{2} \rho_o A_o \{\dot{u}\}^T \{\dot{u}\} + \rho_o A_o \{\bar{X}+u\}^T \{f\} +$$

$$+ \{\bar{X}+u\}^T \{W\} \{X'+u'\} + \{X+u\}^T \{F_\alpha\} \bar{\delta} \ (S_o - S_\alpha) -$$

$$- \frac{1}{2} E A_o \left[1-(\{X'+u'\}^T \{X'+u'\})^{1/2}\right] dS_o dt \qquad (21)$$

If the last term of the functional is developed in series and only the linear and quadratic terms of $\{u'\}$ are taken into account, considering small movements, the result is

$$J_p = \int_{t_1}^{t_2} \int_{o}^{l_o} \frac{1}{2} \rho_o A_o \{\dot{u}\}^T \{u\} + \rho_o A_o \{\bar{X}+u\}^T \{f\} +$$

$$+ \{\bar{X}+u\}^T [W] \{X'+u'\} + \{X+u\}^T \{F_\alpha\} \bar{\delta} \ (S_o - S_\alpha) -$$

$$- \frac{1}{2} E A_o \left[1+\{X'\}\{X'\}^T+\{u'\}^T\{u'\}+ 2\{X'\}^T\{u'\}-\right.$$

$$\left. - 2(\{X'\}^T\{X'\})^{1/2} - \frac{2\{X'\}^T\{u'\}}{(\{X'\}^T\{X'\})^{1/2}} - \frac{\{u'\}^T\{u'\}}{(\{X'\}^T\{X'\})^{1/2}} + \right.$$

$$\left. + \frac{(\{X'\}^T\{u'\})^2}{(\{X'\}^T\{X'\}^{3/2}} \right] \Bigg\} dS_o dt \qquad (22)$$

The functional will now be discretized in the same way as in the static problem, and we will accept that in the finite element:

$$\{u_i(S_o,t)\} = [N_i(S_o)] \{U_i(t)\} \qquad (23)$$

N_i being the same matrix as that which appears in Equation 6.

By carrying out the appropriate operations the following equation is obtained:

$$\bar{J}_p = \sum_{i=1}^{n} \int_{t_1}^{t_2} \int_{S_i}^{S_{i+1}} \frac{1}{2} \rho_o A_o \{\dot{U}_i\}^T [N_i]^T [N_i] \{\dot{U}_i\} +$$

$$+\rho_o A_o \{X_i \; U_i\}^T [N_i] \{f\} + \{X_i + U_i\}^T [N_i]^T [w] [N_i'] \{X_i + U_i\} +$$

$$+\{X_i + U_i\}^T [N_i]^T \{F_\alpha\} \bar{\delta} (s_o - s_\alpha) - \frac{1}{2} E A_o \Bigg[1 + \{X_1\}^T [N_i']^T \cdot$$

$$\cdot [N_i'] \{X_i\} + \{U_i\}^T [N_i'] \{U_i\} + 2\{X_i\}^T [N_i']^T [N_i'] \{U_i\} -$$

$$- 2(\{X_i\}^T [N_i']^T [N_i'] \{X_i\})^{1/2} - \frac{2\{X_i\}^T [N_i]^T [N_i'] \{U_i\}}{(\{X_i\}^T [N_i']^T [N_i'] \{X_i\}^{1/2})} -$$

$$- \frac{\{U_i\}^T [N_i']^T [N_i'] \{U_i\}}{(\{X_i\}^T [N_i']^T [N_i'] \{X_i\})} + \frac{(\{x_i\}^T [N_i']^T [N_i'] \{U_i\})^2}{(\{X_i\}^T [N_i']^T [N_i'] \{X_i\})^{3/2}} \Bigg] \cdot$$

$$\cdot \, ds_o \, dt \qquad (24)$$

Consequently:

$$\overline{\delta J} = \sum_{i=1}^{n} \{U_i\}^T \int_{S_i}^{S_{i+1}} -\rho_o A_o [N_i]^T [N_i] \{\ddot{U}_i\} +$$

$$+\rho_o A_o [N_i]^T \{f\} + [N_i']^T [N_i'] [w] \{\mathbf{x}_i\} + [N_i]^T \{F_\alpha\} \bar{\delta} (S_o - S_\alpha) -$$

$$- E A_o \Bigg[[N_i']^T [N_i'] \{U_i\} + [N_i']^T [N_i] \{\mathbf{x}_i\} -$$

$$- \frac{[N_i']^T[N_i']\{X_i\}}{(\{X_i\}^T[N_i']^T[N_i']\{X_i\})^{1/2}} - \frac{[N_i']^T[N_i']}{(\{X_i\}^T[N_i'][N_i']\{X_i\})^{1/2}}\{U_i\} +$$

$$+ \frac{[N_i']^T[N_i']\{X_i\}\{X_i\}^T[N_i']^T[N_i']}{(\{X_i\}^T[N_i']^T[N_i']\{X_i\})^{3/2}}\{U_i\}\Bigg|\Bigg\} dS_o\, dt \quad (25)$$

which can also be written in the following way:

$$\delta\overline{J_p} = \int_{t_1}^{t_2} \{\delta U\}^T \left[-[M]\{\ddot{U}\} + \{F_m\} + [K_w]\{X_\tau\} + \{F_p\} - \right.$$

$$\left. -[K_L]\{U\} - [K_L]\{X\} - [K_T]\{X\} - [K_T]\{U\} - [K_r]\{U\}\right]\Bigg\} dt \quad (26)$$

where

$$[M] = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} \rho_o A_o [N_i]^T[N_i]\, dS_o \quad (27)$$

$$[K] = \sum_{i=1}^{n} \int_{S_i}^{S_{i+1}} E\, A_o \frac{[N_i']^T[N_i']\{X_i\}\{X_i\}^T[N_i']^T[N_i']}{(\{X_i\}^T[N_i']^T[N_i']\{X_i\})^{3/2}} dS_o \quad (28)$$

From Equation 26 with the necessary condition that $\delta\bar{J} = o$ and, keeping in mind the boundary conditions and Equation 17, the following is obtained;

$$[M]\{\ddot{U}\} + ([K_L] + [K_T] + [K_r])\{U\} = \{0\} \quad (29)$$

that is the equation of the small movements. It is, obviously, a linear equation. Its solution may be sought in the form:

$$\{\ddot{U}\} = \{a\}\, e^{i\omega t} \quad (30)$$

Substituting this value in Equation 18:

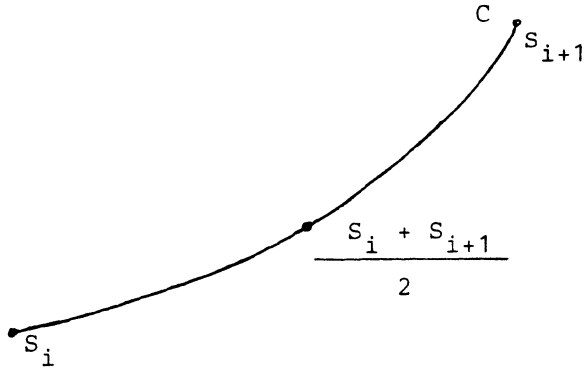
$$([K_L] + [K_T] + [K] - w'[M])\{a\} = \{0\} \qquad (31)$$

a linear equation in (a) compability of which requires the annulation of the determinant of the matrix which multiplies (a), giving as a result, the natural frequencies of the system and, in their function, the eigenvectors.

In the examples presented in this article, the frequencies and eigenvectors were obtained from Equation 31 by means of the combined inverse iteration-bisection method. The Gram-Schmidt orthogonalization method was used to separate multiple frequencies in the inverse iteration.

FINITE ELEMENT USED

For most of the problems concerned with cables or networks which lack stiffness on flexion, the three-node finite element (Fig 1) is very useful and makes it possible to use parabolic interpolation functions. This element, which does not ensure



the continuity of the derivative in the curve at the end nodes but which does so along the finite element, is well suited to the discontinuities imposed by concentrated loads. For this reason, it is necessary for these always to act on end nodes of finite elements. In addition, this method avoids calculation of the equivalent nodal forces, Equation 13.

In order to make operations easier an adimensional variable ξ has been used on every finite element:

$$\xi = \frac{S_o - \frac{S_i + S_{i+1}}{2}}{\frac{S_{i+1} - S_i}{2}} \qquad (32)$$

and, therefore, the integrals are expressed in the following way:

$$\int_{S_i}^{S_{i+1}} \phi(S_o)\, dS_o = \frac{S_{i+1} - S_i}{2} \int_{-2}^{1} \Phi(\xi)\, d\xi \qquad (33)$$

METHOD TO CALCULATE THE STATIC EQUILIBRIUM CONFIGURATION AUTOMATIC SELECTION OF AN INITIAL CONFIGURATION

The non-linearity of Equation 17 - matrix $|K_T|$ depends upon $\{X\}$ - requires the selection of an iterative method which assures convergence. Amongst those tested, the Newton Raphson method was the most effective, as suggested by W.M.Henghold and J.J. Russell (Henghold y Russell, 1976).

Assuming an initial deformed configuration $\{X_o\}$, the additional loading system $\{P_o\}$ can be calculated so that (X_o) effectively becomes the static equilibrium configuration.

That is:

$$\{P_o\} = ([K_L] + [K_T])\{X_o\} - \{F_m\} - \{F_p\} - [K_W]\{X_o\} \qquad (34)$$

Next, the variation $\{\delta X_o\}$ must be calculated in such a way that:

$$\{X\} = \{X_o\} + \{\delta X_o\} \qquad (35)$$

is the solution to Equation 10.

Substituting Equation 35 into Equation 17 gives:

$$([K_L] + [K_T(X_o + \delta X_o)])(\{X_o\} + \{X_o\}) = \{F_m\} + \{F_p\} + [K_W](\{X_o\} + \{X_o\}) \qquad (36)$$

so

$$([K_L]+[K_T(X_o)])\{X_o\}+([K_L]+[K_T(X_o)])\{X_o\}+$$

$$+[\delta K_T](\{X_o\}+\{X_o\}) =\{F_m\}+\{F_p\}+[K_w](\{X_o\}+\{X_o\}) \quad (37)$$

In a first iteration it can be supposed that $\{\delta X_o\}$ is small. Therefore, by developping the preceding expression, eliminating superior terms and taking into account Equation 34:

$$(-[K_w]+[K_L]+[K_T(X_o)]\{X_o\} + [\delta K_T]\{X_o\} = -\{P_o\} \quad (38)$$

But

$$[\delta K_T]\{X_o\} = [K_r]\{\delta X_o\} \quad (39)$$

as it can be proved from the very definition of K_r and K_T:

$$([K_L] - [K_w] + [K_T(X_o)]+[K_r(X_o)])\{\delta X_o\} = -\{P_o\} \quad (40)$$

Once $\{\delta X_o\}$ has been obtained, we proceed iteratively with a new value:

$$\{X_1\} = \{X_o\} + \{\delta X_o\} \quad (41)$$

which leads to a new system of additional loads $\{P_1\}$ etc ....

The number of necessary iterations can be limited by restricting the maximum relative error

$$\frac{\{\delta X_i\}}{\{X_i\}}$$

as suggested by W.M.Henghold and J.J.Russell.

When the final configuration is known, the tension on any point of finite element τ can be calculated, starting from the following equation:

$$\sigma = E\left[\left(\{X_\tau\}^T[N'_\tau]^T[N'_\tau]\ \{X_\tau\}\right)^{1/2} - 1\right] \qquad (42)$$

The value of σ obtained this way is subject to significant errors if the deformation is small.

The iterative method explained requires knowledge of the initial configuration $\{X_o\}$. When the problem refers to a single cable, it is easy to establish a reasonable initial configuration $\{X_o\}$ by a simple manual calculation. But in the case of spatial networks with a large number of cables, this operation can be very long and tedious. In addition, it is quite possible that the initial configuration thus obtained be very far from the real equilibrium configuration, which would increase the number of steps in the iterative process.

For this reason, the following procedure is proposed for the automatic selection of the initial configuration $\{X_o\}$.

In the first place the equation is solved:

$$[K_L]\{X_t\} = \{P_o\} \qquad (43)$$

$\{X_T\}$ is a vector which contains the pre-established nodal co-ordinates - natural boundary conditions - while the remaining coordinates are considered as unknown quantities.

$\{P_o\}$ is the nodal forces vector, zero in the ordinary nodes and having unknown values - which are not zero - in the nodes with pre-established co-ordinates

$\{X_t\}$ can be calculated from Equation 43 and, with this vector, the following can now be calculated:

$$\{P\} = [K_w]\{X_t\} + \{F_m\} + \{F_p\} \qquad (44)$$

Next {Xp} is calculated from the equation:

$$[K_L]\{X_p\} = \{P\}$$

The differential vector

$$\{\Delta\} = \{X_p\} - \{X_t\} \qquad (46)$$

is a nodal displacement vector which is logically related to the system of loads applied to the network or cable.

The configuration $\{X_o\}$ was obtained by means of the expression:

$$\{X_o\} = \{X_t\} + \lambda\{\Delta\} \qquad (47)$$

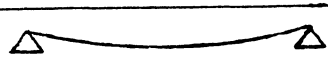
where λ is a numerical coefficient chosen so that the length of the cable or the total length of the network in the configuration $\{X_o\}$ be a pre-established percentage superior to that of the configuration without deformation.

It is obvious that the method for selection of $\{X_o\}$ can give rise to initial configurations relatively far from the final solution - especially when the cables or networks present great variations in the slopes or loads which depend upon the position. Nevertheless, the manual selection of $\{X_o\}$ does not avoid this risk in the slightest.

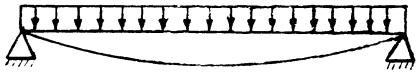
It has been experimentally proved - at least in all the examples solved - that the Newton Raphson method always converges on the final solution when starting from the $\{X_o\}$ obtained by this automatic procedure.

COMPARISON OF RESULTS

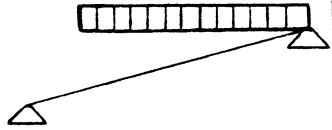
1. Cable of 491.02m in length; weight 5'28 KN/m suspended between two points placed at the same height - span of 488m, subject only to the force of its own weight.

	Sag in the centre (m)
Wilson and Wheen /3/	24.4
This study (10 elements)	24.4515

2. Cable of 491.02m in length; weight 5.28 KN/m, suspended between two points placed at the same height - span 488m - subject to the action of its weight and to the uniform load along the span of 17.52 KN/m.

	Sag in the centre (m)	Supporting Angles (9.sexog)
Wilson and Wheen /3/ (1 segment)	26.731	12.559
Wilson and Wheen /3/ (2 seconds)	26.769	12.376
Wilson and Wheen /3/ (3 seconds)	26.765	12.400
Wilson and Wheen /3/ (1 segment. met perf)	26.756	12.371
Harrison /9/ (580 elements)	26.785	12.407
This study (10 elements)	26.839	12.387

3. Cable of 485.25m in length, wight 5,28kN/m, suspended between two points placed at different heights (difference in height 12,2m, span 488m) and with a uniformly distributed load to the right half of the span of 17.52KN/m.

	Sag IN THE CENTRE (m)	
	WEIGHT ONLY	WEIGHT + LOADING
Wilson and Wheen /3/	7.943	10.835
This study	7.938	10.817

4. Cable of 313.047m in length, weight 46KN/m, suspended between two points placed at the same height - span 305m - with a concentrated load, P, of 35'6KN applied to a distance of 122m from one of the ends. $A_o = 5.5 \times 10^{-4} m^2$.

Every author mentioned in these four examples considers the weight of the cable as a constant load per unit of span. This fact, which in every case leads to underestimating its effect, partly justifies the divergence among these results and those obtained by the authors of this study.

5. Calculation of the frequencies of a stressed network with eight supporting points. Tension per cable 334Kw, span 3.1m.

Cuerda vibrante equivalente.- Equivalent vibrating cord.
Primera frecuencia natural.- First natural frequency.
Análisis de masas concentradas.- Concentrated mass analysis.
Analogía de la membrana.- Analogy of the membrane.

CONCLUSION

The variational principles presented allow for the study of the static and dynamic behaviour of cables and networks subject to the action of general load systems. Application of the F.E.M. to the solution of these problems leads to results which are in agreement with those obtained by several authors using other methods, surpassing these latter in generality since the load system, as indicated, can be made as complex as necessary.

Automatic selection of the first iteration position is an advantage as it eliminates bothersome and complex calculations which, in many cases, may lead to non-convergence of the programme. Besides, the functional defined can be adapted to the study of cases and networks with stiffness on flexion, and internal and external attenuation.

REFERENCES

Bastero, C. (1980) Un principio variacional para la dinámica de continuos unidimensionales elásticos. Doctoral Thesis, San Sebastián.

Gambhir, M.L. and Batchelor, B. de V (1977) A Finite Element for 3-D Prestressed Cablenets. Int. Jour. for Num. Meth. in Eng., Vol. II, pp. 1699-1718.

Harrison, H.B. (1970) Suspension Cable Movement Analysis. Civil Engineering Transactions, Institution of Engineers, Australia, April, pp. 72-78.

Henghold, W.M. and Russell, J.J. (1976) Equilibrium and Natural Frequencies of Cable Structures (a non-lineal finite element approach). Journal of Computers and Structures, Vol. 6, pp. 267-271.

Leonard, J.W. (1973) Incremental Response of 3-D Cable Networks, Journ. of the Engin. Mech. Div., Proc. ASCE, June, pp. 621-629.

Michalos, J. and Birnstiel, C. (1962) Movements of a Cable Due to Changes in Loading", Transactions ASCE, Vol. 127, paper no. 3368, pp. 267-303.

O'Brien, W.T. and Francis, A.J. (1964) Cable Movements under Two-Dimensional Loads", Journal of Structural Division, Proc. ASCE, Vol. 90, June, pp. 89-123.

Peyrot, A.H. and Goulois, A.M. (1979) Analysis of Cable Structures", Computers and Structures, Vol. 10, pp. 805-813.

Pugsley, A.G. (1957) The Theory of Suspension Bridge", Edward Arnold, London.

Routh, E.J. (1891) Analytical Statics", Cambridge, Vol. 1, chapter X.

Sangster, K.G. and Batchelor, B. de V. (1974) Non-linear Dynamic Analysis of 3-D Cablenets". Computational Methods in Non-linear Mechanics, Proc. Int. Conf. Comp. Meth. Non-linear Mech., Univ. of Texas at Austin, pp, 301-310.

Shore, A. and Chaudhari, B. (1972) Free Vibration Cable Network Utilizing Analogons Membrane. Proc. Int. Assoc. Bridges and Struc. Eng., Ninth Congress, Amsterdam, pp. 445-451.

Soler, A.J. and Afshari, H. (1970) On Analysis of Cable Network Vibrations using Galerkin Method, Jour. of App. Mechan. Trans. ASME, September, pp. 606-611.

Wilson, A.J. and Wheen, R.J. (1977) Inclined Cables under load-design expressions". Jour. of Struc.Div., Proc. ASCE, May, pp. 1061-1078.

CAD/CAM

COMPUTER AIDED DESIGN OF INTERIOR ARTIFICIAL LIGHTING OF BUILDINGS.

D.J.Carter, Muspratt Laboratories, Liverpool, England

INTRODUCTION

In designing artificial lighting for a building interior, designers have in the past concentrated on the provision of planar illuminance, usually on some horizontal working surface, together with a check on the likelihood of direct discomfort glare from the proposed installation. The design tools most usually used for this purpose have been the lumen design method (1) coupled with a glare index calculation (2), although these are strictly speaking methods of appraisal rather than methods of "design". This process has the advantage of being economical in terms of manual design time, but it ignores other aspects of lighting design which may influence the quality and appearance of the finished lighting scheme. The popularity of the lumen design/glare index check method also extends to computer aided design. This may be due to the absence of economical design methods which apply the quality criteria to lighting design.

The quality criteria of lighting design have been the subject of research effort over the past decade and some quantitative measures for these criteria have now been proposed. An exhaustive list of criteria which could be considered by designers is given in Table 1 (3) although the practical applications of criteria 1, 3 and 10 have not been fully developed.

In ideal conditions lighting design methods would consider all the criteria listed in Table 1. This is not practical for two reasons. Firstly the measures taken to satisfy one criteria may react with others so that under most practical design conditions it is not possible to achieve satisfactory values for all the criteria. Secondly, the amount of calculation required to design an installation to satisfy all the criteria is, even using computers, daunting.

Table 1 : Criteria and Measures

Criterion	Measure
1. Illuminance of a specific object	Illuminance at a point for a plane of appropriate orientation of average illuminance of object's surface.
2. Average illuminance of horizontal surfaces at reference plane	Average horizontal illuminance
3. Average illuminance of vertical surfaces at reference plane	Average mean cylindrical illuminance
4. Average illuminance of surfaces of all orientation at reference height	Average scalar illuminance
5. Evenness of lighting	Uniformity ratio
6. Direct discomfort glare	IES Glare Index
7. Direction of flow of light	Illumination vector direction
8. Directional strength of lighting	Vector/scalar ratio
9. Brightness patterns	Illuminance ratios and reflectances of bounding and other surfaces
10. Emphasis	Items which are to be emphasised should generally have a higher illuminance or luminance than others

The relative importance of the different criteria in any design case depends on the required characteristics of the particular installation and the decision as to which of the possibly conflicting criteria to apply must be made by the designer. This decision is restricted by the design methods that are available. The lumen method/glare index check can only satisfy the criteria of average horizontal illuminance, uniformity ratio and glare index, although a further calculation to produce a crude luminance pattern has been put forward (4).

This paper describes two computer aided solutions that synthesise solutions based on the quality criteria of design, namely, Luminance Design and Multiple Criterion Design. The luminance method (5) is based on knowledge of a given brightness pattern on the principal room surfaces, and a luminaire specification is derived to achieve this pattern. Although the method accurately specifies brightness patterns and average horizontal illuminance, the other design criteria are not considered. The multiple criterion design method (MCD) was first introduced in 1977 and enables the designer to control the properties of a lighting installation in considerable detail and to ensure compliance with a majority of the criteria outlined in Table 1.

PREVIOUS WORK

Although accounts of several programs for daylight prediction have been published, including an excellent comparison of a number of such programs (6), the amount of published material on computer-aided design of interior artificial lighting remains modest. Simons (7) in a review of the use of computers in practice in building services claims that the main application of computers in lighting is for production of photometric data including tables of utilance factors glare indices, and aspect factors. With this in mind the recently published CIBS Technical Memoranda No.5 (8) has been structured so that programmers with only a passing knowledge of lighting terminology are able to write a program to calculate utilization factors. This information is then used in the manual lumen design/glare index check method.

Computer programs have been used by designers as a means of appraisal of proposed or existing installations. Plant and Archer (9) describe a program to calculate illuminance contours on horizontal planes in a rectangular room from both natural or artificial light sources. Information from the program abstract service of organisations such as the National Computer Centre show that a small number of computer bureaux and consulting engineers maintain appraisal programs for point by point illuminance calculations and lumen design. Commercial interest, however, restricts comprehensive publication of these programs.

In contrast to the dearth of published work in lighting design a recent report (10) has shown that micro-computers are used extensively by designers in other construction disciplines, including building services. It is thus not unreasonable to assume that if suitable programs were available, the experience of the present users of building services computer programs could be extended to include lighting design.

MANUAL DESIGN METHODS FOR LIGHTING QUALITY

Luminance design

The manual method of luminance design (5) makes possible the specification of a lighting installation to satisfy the lighting quality criterion of luminance pattern and working plane illuminance only.

The calculation procedure falls into three parts, namely, indirect flux, luminaire specification and downward flux. The procedure may be briefly summarised as follows. From a knowledge of the geometry and desired luminance of each room surface, the amount of indirect light is calculated by reference to tabulated values of inter-reflection factions. The direct illuminance on each surface is then calculated and used to derive the direct ratio and flux fraction ratio for the installation. This enables the choice of suitable luminaire to be made. Finally using the details of the chosen luminaire and the room geometry the total installed flux required is calculated.

The manual design methods for artificial lighting design which satisfy multiple designs criteria are based on work by Jay (11) who showed that it is possible in working environments to find a limited range of conditions in which the majority of the criteria listed in Table 1, can be satisfied simultaneously. This is the case in lighting installations which comprise a regular array of more than four ceiling mounted luminaires installed in a rectangular room having light coloured surfaces and where the lighting has been designed in accordance with the IES Code Recommendations for horizontal illuminance, uniformity ratio and glare index (12). The manual design method put forward by Cuttle (13) makes possible the specification of an installation for normal working environments where the working plane is horizontal and where a regular pattern of fittings on a ceiling is an appropriate method of lighting. The Multiple Criterion Design method (3) described in this section is an extension and refinement of Cuttle's method.

The design method is based on a series of charts (known as IR Charts) relating variations of wall to working plane, and ceiling to working plane, illuminance ratio together with vector/scalar ratio to room differences and luminaire characteristics. Further charts give information on flux utilance (U charts) and glare index (G charts). The IR charts are based on the concept of the luminaire domain (14) in which it is shown that for a given room index and reflectance combination, values of illuminance ratio are represented by straight lines on a linear plot of direct ratio against flux fraction ratio. It was further demonstrated (15) that for practical purposes the locus of vector/scalar ratio will also be a straight line when plotted in a similar manner. The IR charts present the above data for luminaires having BZ classifications

1 to 10 and for twelve combinations of surface reflection factor. The specification of the surface is done by designating each room surface by a letter L (Light), M (Medium), or D (Dark), each letter corresponding to some numerical value of reflection factor.

The U charts are a convenient method of summarising standard upper and lower flux utilance data (16) for all BZ classifications and room sizes.

An installation may be checked for discomfort glare using the IES Glare Index System. The resulting glare index may then be compared with the Limiting Glare Index values appropriate to the installation. Limiting Glare Index represents the maximum permissible value of glare index set out in the IES Code for a given situation. The use of the Glare Index System is a time consuming process due to the large number of variables. A simplified version (17) has been proposed and this forms the basis of the G charts. Glare index values are read from the appropriate G chart for the variables of BZ class, flux fraction ratio, downward flux and luminaire mounting height assuming a standard value of luminous area. The G chart method has an admitted tendancy to trade convenience for accuracy. However, it is claimed that the method is acceptable given the insensitivity of persons to small glare differences.

It is difficult to estimate the amount of design time taken in use of the two manual design methods, since this will vary with such factors as the experience of the designer, his familiarity with the methods, the number of criteria to be satisfied and the number of other environmental engineering factors which will influence the lighting design. Since its introduction in 1977, the manual multiple criteria design method has not been widely used. Price (18) suggests that the main reason for this is that the process is too time consuming for the practicing engineers.

DESCRIPTION OF PROGRAMS

The programs are written in Basic for interactive use on micro computers, the designer providing design information as program input when prompted by the computer. A flow diagram for the luminance design program is given in Figure 1, and for the MCD program in Figure 2.

The programs are applicable to rooms in the form of rectangular prisms and bounded by surfaces which approximate to diffusing reflectors. It is assumed that the proposed artificial lighting installations will consist of a conventional array of regularly spaced luminaires. The programs do not make allowances for any natural light provided by

windows or roof lights. Values for illuminances and luminance have been averaged over the full area of each wall, working plane or ceiling surface. The Luminance Design program is split into three stages. At the start of each stage the program reads in room details and design criteria. Using the desired luminances and reflectances of the walls, ceiling and working plane or floor as a starting point, the first stage program computes the total, indirect and direct illuminances which will be required on each of the surfaces if the desired luminance pattern is to be achieved. The designer must now specify the mounting height of the luminaire and the program will yield the direct ratio, ceiling ratio and flux fraction for an appropriate installation. This information may be simply translated into a luminaire specification of British zonal classification (BZ) upward and downward light output ratios (ULOR and DLOR) (8). It is the task of the designer to select a luminaire with properties close to the ideal specification.

The second stage program will compute the total installed flux and the minimum number of luminaires of the chosen type to satisfy the appropriate spacing/mounting height ratio (Table 2)

Table 2 : Spacing mounting height ratios

BZ Classification	Ratio
1 and 2	1
3 and 4	1.25
5 to 10	1.5

The designer must now choose the lamps and arrange the luminaires in a suitable layout. In many ways this stage is very similar to the lumen method.

The third stage program allows the designer to appraise the current solution in terms of the original specification.

The geometric and photometric properties of the chosen installation act as inputs to an iterative process which will converge on the average illuminances and luminances for each of the six room surfaces. Since the layout of the luminaires is assumed to be regular and the effects of asymmetry are excluded then the direct illuminance on the four walls will be computed as equal. This may conflict with the output of illuminances from the first stage in cases where the original specification included walls differing in luminances.

The Multiple Criterion Design program initially reads in room details and design criteria. If the illuminance criterion is specified as scalar corresponding value of horizontal illuminance is derived. Using the horizontal illuminance criteria,

the program computes the total, indirect and direct illuminance which will be required on each of the surfaces if the desired illuminance ratios are to be achieved. The analysis of the illuminance distribution is achieved by a series of inter-reflection calculations using surface luminance and form factors. Form factors are calculated from the geometry of the room enclosure. These are dimensionless quantities expressing the relative proportions of a surface 'seen' by another surface in the enclosure. Four expressions were derived to define the form factors and incorporated in the program. The inter reflection calculations incorporate Hisano's approximation (19) which places a restriction on the room dimensions to the effect that the ratio of length to width must not be greater than 4 and the room index not less than 0.25.

The program calculates flux fraction ratio and direct ratio by consideration of room geometry and illuminances and then derives the required luminance specification and layout. This is readily accomplised since the BZ classification system is based on the relationship between direct ratio and room index, and each BZ class has an appropriate spacing/mounting height ratio to satisfy the criteria of uniformity ratio. To complete the luminaire specification the downward flux/luminaire is derived in terms of upper and lower flux utilance.

The next stage of the program is the calculation of the actual vector/scalar ratio achieved by the installation. Vector illuminance is calculated assuming that the vector direction is vertically downward from the array of overhead sources and that the illuminance in the upward direction is a function of the horizontal illuminance and the reflection properties of the floor cavity. Scalar illuminance is derived from its direct and indirect scalar components. The direct component is derived using the concept of scalar direct ratio and the indirect components using the B.R.E. Split Flux Method (20).

In the treatment of glare the MCD program differs radically from the manual methods of calculation. The MCD glare calculation directly applies the BRE Glare formula (21) to each luminaire assuming an observer viewing down the longest dimension of the room. The parameters of source brightness, solid angle subtended at observer, position index, and background brightness are calculated for each luminaire in turn, and combined to give the glare index for the whole installation based on a standard luminous area. This method thus eliminates any inaccuracy due to interpolation in graphs or tables which may occur in manual methods. Finally, the program calculates the luminous area correction required for each luminaire such that the limiting glare index criteria is satisfied.

PROGRAM APPLICATION

The use of the programs are demonstrated by an example, comparing the computed results with photometric measurements from an actual installation.

Photometric measurements were made in a conference room and the results used as the basis for input data for the program. Reflection factor was measured using an EEL "Unigalvo" galvonometer and reflectance head. A Hagner Universal photometer was used for luminance and illuminance measurements and vector/scalar ratio was measured using a Megatron Spatial Illuminance Meter. The measured values averaged over each surface are given in Table 3.

Table 3 : Photometric data for Conference Room

Surface	Total Illuminance (lux)	Reflection Factor	Calculated Luminance (Cd/m^2)
Wall 1 Short	184	0.54	31.4
Wall 2 Short	184	0.75	43.5
Wall 3 Long	162	0.49	25.3
Wall 4 Long	150	0.71	33.8
Ceiling	-	0.87	30.0
Working Plane	499	0.22	35.0

Illuminance ratios calculated from the measured values for working plane illuminance and averaged measured wall illuminance are shown in the list of criteria in Table 4. The limiting glare index criteria was taken as the glare index for the actual installation so that comparison between the actual and computed luminaire specification may be valid. It should be noted that the illuminance ratio criteria for the Conference Room are outside the recommendations of the IES Code indicating that the designer of the scheme wished to emphasise the horizontal plane at the expense of the walls.

Table 4 : Design Criteria for Conference Room

Criteria	
Vector/scalar ratio	Not measured
Wall illuminance ratio	0.34
Ceiling illuminance ratio	0.22
Working plane illuminance	500 lux
Limiting glare index	19

The room is 9.8m long, 7.28m wide, 3.0m high and the working plane is a desk top 0.7m above the floor. The room is lighted by twenty four Merchant Adventurers recessed louvered luminaires with vertical 200W GLS lamp and reflector. The fittings have a BZ class of 1, light output ratios of 0% up and 55% down and a luminous area of $0.02m^2$. A plan of the conference room is shown in Figure 3.

The MCD program input/output for this example is shown in Figure 4. The figures in square brackets are design information entered by the user. Input/output for stages One and Two of the Luminance Design Program are shown in Figure 5.

A comparison between actual and calculated luminaire specification for both programs is made in Table 5. The spacing of the luminaires in the actual installation is slightly non-standard and allowance must be made for this before valid comparison can be made with the results of the program in which standard luminaire spacing is assumed. The actual spacing

Table 5 : Comparison of Computed and Actual Installation Specifications.

	Actual Installation	Computed Results	
		Luminance Design	MCD
Flux Fraction Ratio	0	0	0.003
Direct Ratio	0.844	0.850	0.843
BZ Class	1	1	1
Total downward flux	36,300lm	43,000lm	39900 lm
Glare Index (See Note (1)	14.3	-	14.9
Luminous area to achieve Limiting Glare Index	$0.02m^2$	-	$0.019m^2$

Note (1) Assuming $0.0645m^2$ luminous area

between the luminaires is 0.72 x Hm (where Hm is the mounting height of the luminaires above the working plane,) the spacing adjacent to wall 3 is 0.52 Hm, of walls 2 and 4 is 0.305 Hm and of wall 1 is 0.39 Hm. As the recommended spacing/mounting height ratio is unity for BZ1, it is apparent that the effect of the actual luminaire spacing will increase direct ratio and that of the wall spacing will reduce direct ratio. The effect of the spacing on direct ratio may be estimated using information from a similar problem (5). The exact direct ratio for BZ1 and room index 1.816 is 0.824 and it is estimated that luminaire spacing will increase direct ratio by 5% and wall spacing will decrease direct ratio by 2.5%. Thus actual

direct ratio 0.844 which compares favourably with the computed figures. Similarly the non-standard spacing, would tend to depress lower flux utilance and increase total installed downward flux when compared with values for the same installations with standard spacings. The program assumes that luminaires are at the recommended spacing/mounting height ratio for the appropriate BZ class (in this case unity), and generates a five by four array of fittings. Although the actual layout is a six by four array, reasonable agreement between total installed flux figures is obtained since both were calculated to give the same required horizontal working plane illuminance. No measurement was made of vector/scalar ratio for the actual installation but the computed figures of 1.39 from the MCD program agrees well with published data (12) for a room with low BZ class luminaire and light coloured floor. Good agreement between computed values of glare and the IES Glare Index Table is obtained and if the actual glare index for the installation of 18.4 is input as "Limiting Glare Index", good agreement is obtained with actual luminous area.

DISCUSSION

The manual luminance design and multiple criterion design methods represent a considerable advance in artificial lighting design. Multiple criterion design in particular brings science to bear on the choice of luminaires for a working environment such that a range of design criteria are satisfied. The manual methods of design are not as widely used as it should be, and the main reason for this appears to be the lengthy and complex nature of the design process. The main advantage of the computer aided approach is that it relieves the designers of the time consuming element of the design process. It is to be hoped that this freedom from the chore of calculation leaves the designer free to explore alternative design solutions and to understand the implications of his original design decisions. In this respect the programs are powerful tools, since they rapidly allow the designer to investigate the effects on installation performance of altering any of his design criteria. The programs break new ground since although the computer is used extensively in the lighting industry for data preparation and scheme appraisal, its use in design has been limited to lumen design methods. Since many practicing designers now have access to computing facilities, it is to be hoped that a computer aided approach will result in an increased use of luminance design and multiple criterion design methods.

REFERENCES

1. Lighting Industry Federation (1977) Interior Lighting Design Handbook, London.
2. Illuminating Engineering Society (1967). Evaluation of discomfort glare: the IES glare index system for artificial lighting installations. Technical Report No. 10. IES London.
3. Illuminating Engineering Society (1977). Multiple criterion design: a design method for interior electric lighting installations. Technical Report No.15. IES London
4. Illuminaging Engineering Society (1969) IES Code. London
5. Coomber, D.C. and Jay, P.A. (1967) A simplified method of calculation for luminance ratio and designed appearance lighting installations: IES monograph No. 10. IES London.
6. Design Office Consortium (1978) Computer programs for daylighting in buildings. Design Office Consortium Evaluation Report No. 3. DOC Cambridge.
7. Simons, R.H. (1976). Use of computers in interior lighting design. Proc. Sym. Computers in Practice for Building Services. IHVE London.
8. Chartered Institution of Building Services (1980). The calculation of utilization factors. Technical Memoranda No.5 CIBS London.
9. Plant, C.G.H. and Archer, D.W. (1973). A computer model for lighting prediction. Build. Sci. 8, 351.
10. Bensasson, S. (1980). Micros in construction, Design and Construction Computing Association, Cambridge.
11. Jay, P.A. (1968). Inter-relationship of the design criteria for lighting installations. Trans. Illum.Eng.Soc.(London) 33, 47.
12. Illuminating Engineering Society (1977) IES Code London
13. Cuttle, C. (1969). Integrated design method. Architects Journal. 30th July.
14. Lynes, J.A.(1975). The concept of the luminaire domain Ltg.Res. & Technology 7, 185.
15. Lynes, J.A. (1975). The luminaire domain and the flow of light. Ltg. Res and Technology 7, 242.
16. Illuminating Engineering Society (1971). The calculation of utilization factors. The BZ method. Technical Report No. 2. IES London.
17. Lowston J.C. (1962). A simplified method of estimating glare index. Light Ltg. 55, 291.
18. Price, W.A.(1979) Towards better lighting design. Building Services. August
19. Hisano, K. (1946) Light flux distribution in a rectangular parallelpiped Illuminating Engineering (USA) 41,(3),232.
20. Cuttle, C. (1971) Lighting patterns and the flow of light. Ltg. Res. and Technology 3, 3.
21. Petherbridge, P. and Hopkinson, R.G. (1950). Discomfort Glare and the lighting of buildings. Trans. Illum. Eng.Soc. (London) 15, 39.

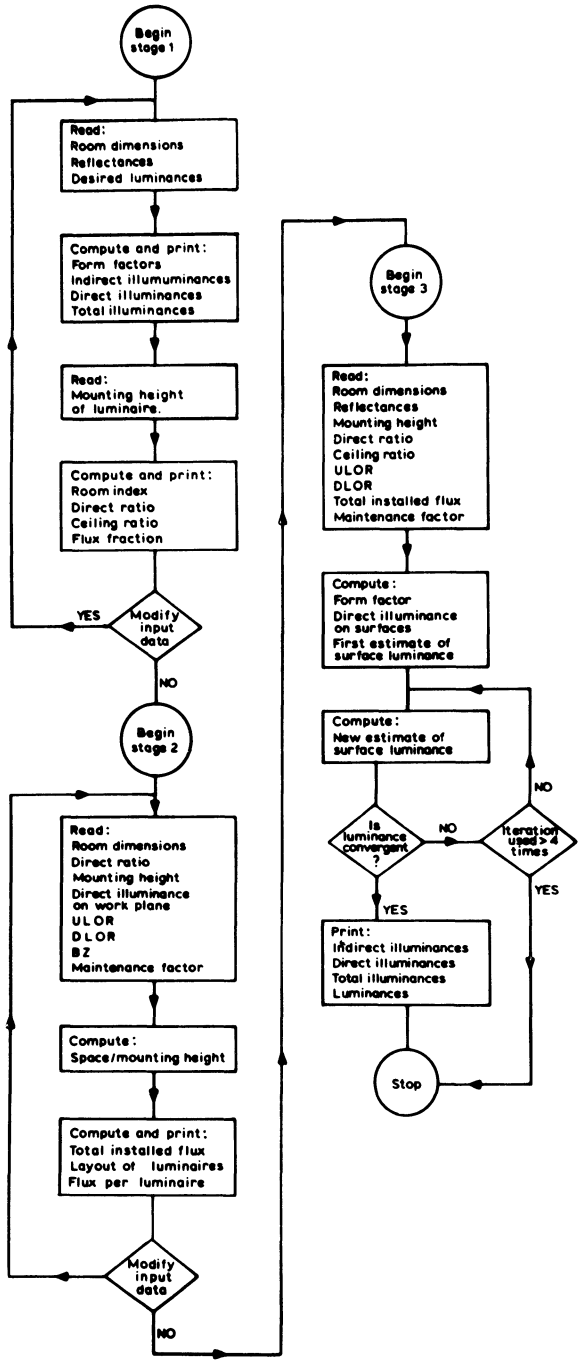


Figure 1 Luminance design flow diagram.

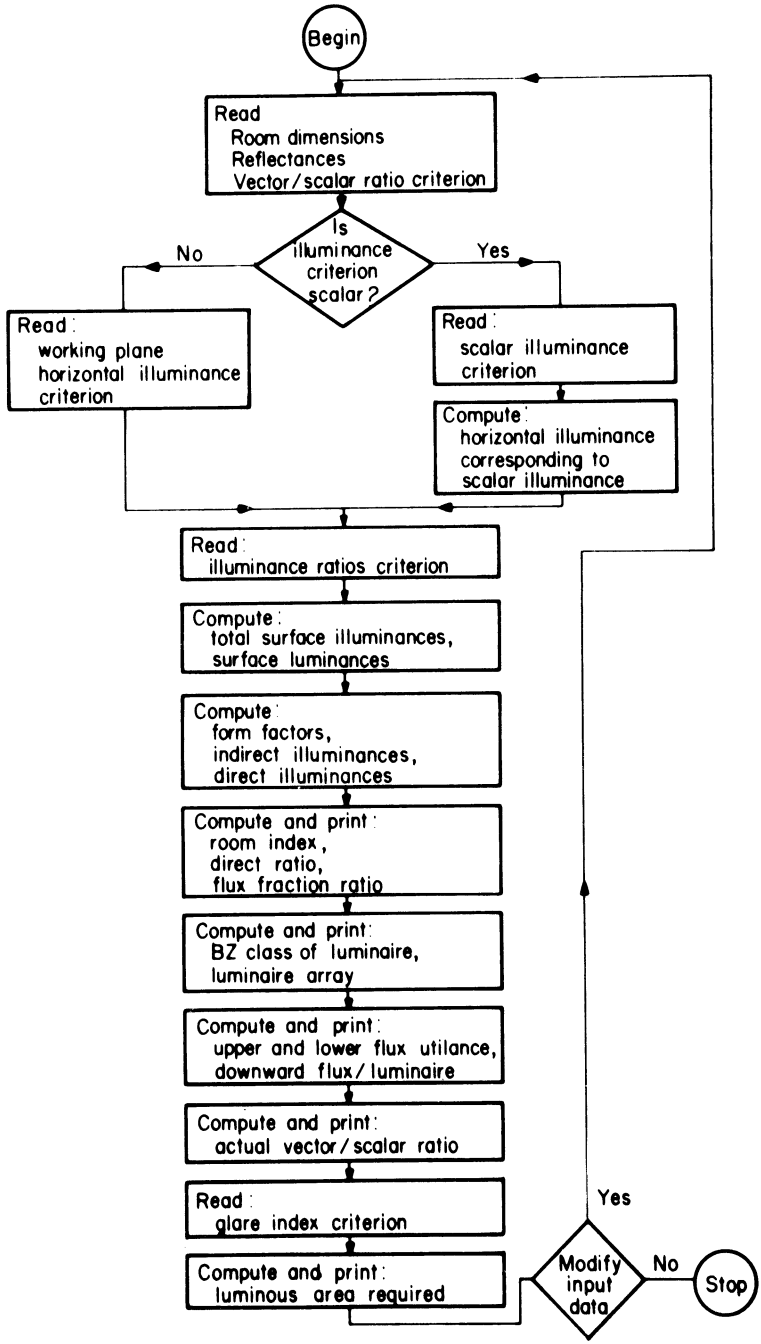


Figure 2 Multiple criterion design flow diagram.

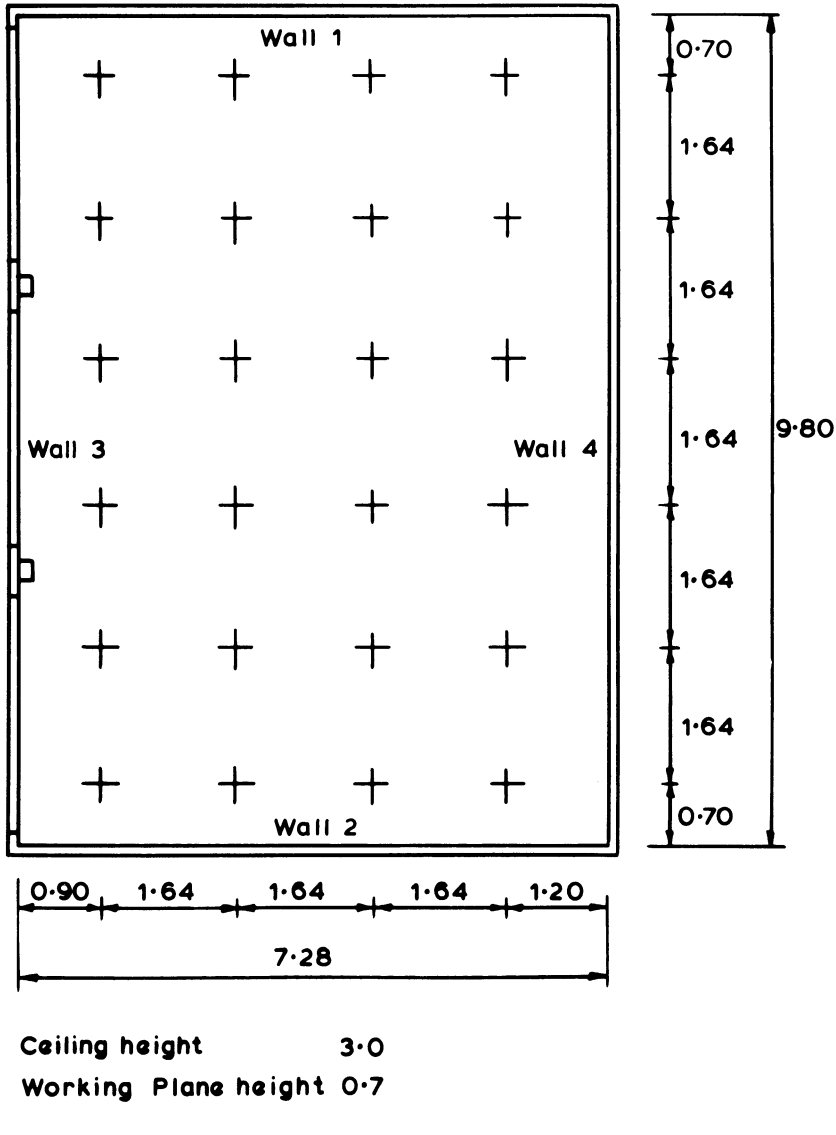


Figure 3 Plan of conference room.

```
MULTIPLE CRITERION DESIGN FOR INTERIOR ELECTRIC LIGHTING
ENTER ALL DIMENSIONS IN METRES
ILLUMINANCE UNITS ARE LUX
LUMINANCE UNITS ARE CANDELAS/SQ.M.
ROOM LENGTH ?                                                    [9.8]
ROOM WIDTH ?                                                     [7.28]
FLOOR/CEILING HEIGHT ?                                           [3.0]
WORKING PLANE HEIGHT ?                                           [0.7]
SUSPENSION LENGTH ?                                              [0]
MOUNTING HEIGHT
2.3
ROOM INDEX
1.81611
REFLECTANCES ?                                                   [0.54, 0.75,
VECTOR/SCALAR ILLUMINATION RATIO PREFERED ?                      [1.5]
TYPE 1 IF ILLUMINANCE IS SPECIFIED AS HORIZONTAL VALUE
 OR 0 IF SPECIFIED AS SCALAR VALUE                               [1]
HORIZONTAL WORKING PLANE ILLUMINANCE ?                           [500]
WALL ILLUMINANCE RATIO                                           [0.34]
CEILING ILLUMINANCE RATIO ?                                      [0.21]
LUMINANCES OF SURFACES
29.2209
40.5845
26.5152
38.42
30.4623
35.0141
ILLUMINANCES OF SURFACES
128.509   92.2507   113.992   78.0076   108.35    99.7016
41.4915   77.7493   54.0081   91.9924   1.65003   400.298
170    170    170    170    110    500
ROOM INDEX
1.81611
DIRECT RATIO
0.843782
CEILING RATIO
1
FLUX FRACTION
0.347807E-2
BZ CLASS
1
LAYOUT OF LUMINAIRES
NUMBER ALONG LENGTH
5
NUMBER ACROSS WIDTH
4
TOTAL NUMBER OF LUMINAIRES
20
LFU
0.899845
UFU
0.649558
DOWNWARD FLUX / LUMINAIRE
1977.16
ACTUAL VECTOR/SCALAR RATIO
1.39497
LIMITING GLARE INDEX ?                                           [18.
BACKGROUND BRIGHTNESS
55.9225
GLARE INDEX
14.8568
LUMINOUS AREA REQUIRED (SQ.M)
0.199012E+1
STOPPED AT LINE 999
```

Figure 4 Input/output for M.C.D. program.

LUMINANCE DESIGN FIRST STAGE
ROOM 1
LENGTH 9.8 WIDTH 7.28 CEILING HT 3.0 WORK HT 0.7

LUMINANCES AND REFLECTANCES

	WALL 1	WALL 2	WALL 3	WALL 4	CEILING	WORK
LUM	31.4	43.5	25.3	33.8	30.0	35.0
REF	0.54	0.75	0.49	0.71	0.87	0.22

ILLUMINANCES

IND	104	100	105	102	108	99
DIR	78	82	57	47		401
TOT	182	182	162	149	108	500

HEIGHT OF FITTINGS ABOVE FLOOR 3.0
INDEX 1.816 DIRECT RATIO 850 CEILING RATIO 1.000 FRACTION .000

LUMINANCE DESIGN SECOND STAGE
ROOM 1

LENGTH 9.8 WIDTH 7.28 FTG 3.0 WORK HT 0.7
DIRECT RATIO 0.824 WORK PLANE ILLUMINANCE 400 MAINT 0.8
FTG REF 1 ULOR 0.0 DLOR 0.55 BZ CLASS 1

INSTALLED FLUX 78711 NUMBER FTGS IN LENGTH 5 IN WIDTH 4
TOTAL NUMBER FTGS 20 FLUX PER FTG 3936

Figure 5 Input/output for luminance design program.

THE SOFTWARE INTERFACE BETWEEN CAD AND CAM

Peter F. McGoldrick

Department of Production Engineering and Production Management,
University of Nottingham, Nottingham, U.K.

SYNOPSIS

This paper addresses itself to this problem interface; an interface which, because of the technical and organisational difficulties involved, has not received the attention it merits.

Currently the only area where any established software exists is at the interface between some CAD systems and the NC machining sector of CAM. It is argued that even in this limited area the approach that has been adopted up till now is not the optimum.

The role of the part-programmer is that of interpreting the designer's requirements as stated on the drawing and converting them into an ordered form for processing and post-processing into the final instruction set used to drive the NC machine tool. This task requires knowledge in two areas; firstly a method of planning the particular process being used for production and secondly a procedure for modelling the path generated by the relative motion between the workpiece and the tool. Algorithms are being developed to enable the first of these requirements to be met and some graphics techniques to give a computer driven solution to the second are described.

This paper argues that the part-programmer's tasks can be replaced by software and that provided the organisational gulf between design and manufacture can be bridged then there is no reason why the designer cannot request the computer to produce an NC tape image which has been fully planned, processed and post-processed entirely automatically. The designer can then visually check that his requirements have been fully met.

Full integration of CAD with CAM demands that such software interfaces are identified and developed.

INTRODUCTION

Even though they may not always acknowledge the fact, the overall aims and objectives of each and all the members of an industrial organisation are the same. Individual and group objectives within that framework can however be very different one from the other. This dichotomy is nowhere more apparent than in the relationship between design and manufacture - the two arms of engineering.

The symbiotic nature of this interaction is such that conflict should not exist; it is after all transparently obvious that a product cannot be manufactured until it has been designed - albeit in an ad hoc fashion - and equally the best design is worthless without the productive capacity to turn it into reality. The purposes, priorities and job tasks of design and production are, at times, so different that, without the departmental, compartmentalised nature of most companies to exacerbate the problem, some abrasion at best is almost inevitable.

The introduction of computer assistance with design and manufacture should ensure that a fundamental look is taken at an enterprise's methods and systems, since it is universally recognised that, without a review of the basics, the new technology will not work to its best, and may even be an economic and commercial failure. This review must recognise that the interface between design and manufacture which hitherto has simply consisted of an engineering drawing becomes much deeper; the experienced, trained eye which can interpret a series of lines, dimensions and tolerances is replaced by a need for compatibility of databases and the system's ability to present the contents of that database to different people in different functional roles in different ways; a problem which is far from being trivial.

DEFINITIONS

To attempt to rigidly define terms in an area of rapidly changing technology is futile; nuances of system design and specification alter and move boundaries so frequently that they can become meaningless. In an attempt however to aid comprehension, some definitions will be attempted.

CAD

This is a term which definitely loses and gains something in mid-Atlantic. In the United States of America the term is usually taken to mean Computer Aided Drafting. Setting aside the possibility that this would translate in the United Kingdom into Computer Aided Draughting, the term in Europe more usually is taken to mean Computer Aided Design. With market domination across the world in the hands of American system purveyors

either interpretation is however equally valid.

Computer Aided Drafting Most of these systems have no pretensions to be anything other than as described by their generic name; they are simply a mechanised system to replace drawing and detailing effort. The application of such systems is justified when there is an existing or anticipated skills shortage, a manpower recruitment problem, in an attempt to increase the productivity of an individual draftsman or when the high quality output produced by sophisticated plotters is perceived as having a marketing advantage.

Such systems are not really of concern here since they are rarely capable of extension beyond the straightforward production of drawings for which they were intended.

Computer Aided Design Much as if one were attempting to describe the difference between a camel and a horse it is difficult to say what it is that distinguishes a design system from a drafting package; but it is easy to recognise it when one sees it!

Design is much more than mere drafting; it is the ability to nest components, to scale them, to alter them rapidly, to recall a library of standard parts and components; it is the ability to doodle intellectually as well as graphically. When a design is acceptable it can be drafted.

CAE

Computer Aided Engineering has been defined as "the step beyond CAD/CAM" (Parmater, 1980). It adds functions which extend the ability of the designer by the creation of a system model in the computer; this model is derived from the results of tests, analyses and data-bank information already stored. CAE can help to replace part of the development cycle - the building and testing of prototypes - by providing the designer with techniques to carry out static and dynamic stress calculations, and to compute natural frequencies and modal shapes; procedures such as finite element methods with automatic mesh generation are typically used.

The involvement of CAE systems with manufacture usually tends to be limited to assistance with the production of prototypes; that part of manufacture which is concerned with the development cycle.

CAM

In its simplest form computer aided manufacture incorporates Numerical Control (N.C.) and its relatives CNC (Computer N.C.) and DNC (Direct N.C.) as well as robotics. In fact the moves by several computer manufacturers to regard machine tools and robots as simply computer terminals seems to be an entirely sensible and logical step; a step which will encourage people

to think of the whole of manufacturing industry as a system to which systems thinking and approaches are appropriate.

CIM

Computer Integrated Manufacturing then becomes the next logical step. As the name implies it seeks to integrate all the business, technical and managerial aspects of the operation of a manufacturing enterprise. The various aspects have been neatly defined (Anonymous, 1982) under the headings of: Business Planning and Support, Engineering Design, Manufacturing Planning, Manufacturing Control, Shop Floor Monitoring and Process Automation. Computer supervision of all these areas with the output of one function serving as input to the next and the sharing of common data between them is the aim. It is not known whether any systems with this totality of approach yet exist.

THE INTEGRATION OF CAD WITH CAM

Conceptually the integration of CAD with manufacture is a small step. It is patently obvious that for example the cartesian coordinates of the centres of a set of holes drilled in a component such as a boiler plate can be used to instruct a drafting machine to draw the holes or an N.C. machine tool to produce them. A sculptured three dimensional surface equally can be defined by a set of digital data which can either instruct a plotter to produce a drawing or an N.C. milling machine to cut metal.

This area of integration is still very weak (Gregory, 1982) and the reasons are, as so often, historical and organisational.

Historical difficulties

CAD systems have been developed in many cases purely for the one purpose. It is not surprising then that the database selected is not suited to the purposes and needs of manufacture.

Where a company has developed a system for its own in-house use and has put it on the market for other organisations to purchase then there is a greater probability that manufacture may have been borne in mind in the specification of that system; conversely of course the flexibility may be reduced to cover only those processes and product types which are common in the parent company.

Organisational problems

There is no doubt that the departmental organisational structure commonly used in productive industry legislates against the integration of design and manufacture - whether computer assistance is used or not. Much of the conflict alluded to in the introduction relates to this factor.

If it were possible to revert to a structure where a group was

responsible for all aspects of a product then many of the problems could disappear along with the "opposing camps" of large design and production departments. It would be an interesting debate to try and judge whether such small product centres could be economically viable.

PRESENT STATE OF THE ART

There are two areas which are critical to the successful integration of CAD with CAM which have received some attention over the last few years.

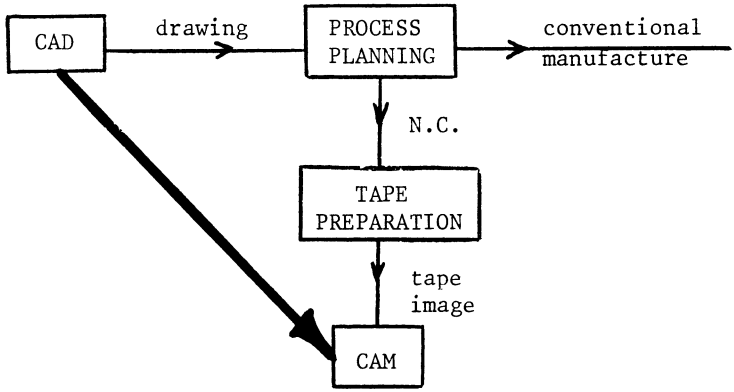


Figure One

As shown by the bold line in Figure One, the link between CAD and CAM must encompass the automatic handling of those routines covered by the functional areas indicated as process planning and tape preparation. These areas have received some attention but, as discussed later, the approaches may not be the optimum.

N.C. Tape Preparation
This is the only area where some success is evident; the input to N.C. machining from some CAD systems is now widely available on a variety of commercial systems.

These systems differ in style, configuration and necessary hardware but their basic principles are very similar. The cutter centre line path is defined by a series of lines, points and circles and the cutter is then "driven" around a path linking various parts of the geometry. The data is then processed and post-processed before producing the control tape or tape image.

Off-line robot programming This is related to the preparation of N.C. tape since it is a similar co-ordinate manipulation exercise. Work is proceeding apace and some promising work

links the GRASP robot simulation package with off-line programming (Bonney et al, 1983).

Process planning

This is that stage when a systematic determination is made of the methods, processes and machines which are to be used to manufacture a product at the required level of quality and economically in the right quantities. It is not a trivial task to devise algorithms which can compute a solution to a problem such as this for which there is no unique optimal answer. A study has been described (Halevi, 1980) in which four process planners produced four different solutions to the operation which simply involved the production of a forty millimetre diameter hole. The possible permutations of feasible solutions to a complex machining problem must be enormous - but so then is the scope for potential improvements.

Much excellent work has been done (O'Brien et al, 1982) but much remains to be done.

ROLE OF THE PART-PROGRAMMER

This is something which will change dramatically with integration; a change which is long overdue.

Traditional role

Ignoring the process planning function which many part-programmers undertake, and which has been discussed earlier, the role is essentially that of a data convertor. The data which is intrinsic in the drawing is converted into a form which is compatible with the input to the processor; its output in turn is transmitted to the post-processor section of the program which then gives a tape or tape image.

Traditionally the role has grown with the technology. When N.C. was first introduced it was so user-unfriendly as to ensure that a role was invented to cope with a task which needed specialist background and training to carry it out. That surely is no longer the case.

The future

In a linked CAD/CAM system with automatic process planning and tape preparation, the part-programmer has no role. It becomes perfectly possible and, from an efficiency point of view, totally desirable to equip the CAD designer with the ability to instruct the system to "prepare for manufacture". This instruction will cause the system to select the production process, plan it, pass the data through the processor and post-processor and present him interactively with an image which he can judge to be right or wrong according to his needs. This image, a picture of the complete manufactured component, must be presented in an understandable fashion. The current plots of the cutter centre line path produced by existing systems will

not be sufficient, and a novel method of coping with this problem is discussed later.

TOWARDS FULL INTEGRATION

CAD and CAM can only be integrated by considering both aspects together right from the outset; an attempt to graft on a manufacturing package to the end of a CA Drafting system will almost certainly fail - at best it could only be achieved at considerable time and software costs.

General packages designed for other companies and their products may not be entirely suited to another's use. Any resulting compromise might result in unacceptable changes to existing systems or corruption of the quality of products or their design which could not be contemplated.

In-house development of software and hardware is the ideal solution and has much to recommend it - but skills shortages and development costs could be decisive factors against.

The hardware blind alley

It is often the case that CAM is viewed as an equipment centred activity. A lot of research effort and money has been devoted to improvements in processes and machines - sometimes at the expense of the total system.

Mechanical and electronic/control engineers find the challenge of a new piece of hardware investigation particularly stimulating; the much greater imagination and foresight required of a production engineer concerned with the totality of a CAM system often seems to be absent - it is certainly often viewed as less rewarding.

It is entirely reasonable to argue that since most manufacturing organisations still do not have the ability to utilise existing CAM systems, the benefits to be gained from fine tuning hardware must, in overall national productivity terms, be small. The benefits from software improvements can however be considerable.

The software challenge

An imaginative approach to the implementation of the software interface between CAD and CAM is required, and careful attention to detail could reap tremendous rewards.

As can be seen by considering the position of tolerances, these benefits can come from unexpected directions. The capability of an automatic machine or process is fixed; unlike its manual counterpart, where a skilled craftsman can improve on the basic process capability at the expense of much increased time and care, it is not possible to improve on the intrinsic accuracy of the machine. Given this fact, the CAD designer should be able to specify the required tolerance on a particular

dimension perhaps using the appropriate Standard Tolerance designation (British Standards Institution, 1969). The computer aided process planning system then examines the capabilities of the various processes available and selects those which are, from the tolerance standpoint, feasible. There is no need to specify the tolerance in detail in the system or on the output - be it screen or hardcopy; indeed if a limited range of processes is available with a fixed accuracy level then tolerances need not be specified at all. Some work has been done to enable data to be collected in this area (McGoldrick and Brinsmead, 1982) but much remains to be achieved.

A further example can be illustrated by comparing the contents of Figure Two with that of Figure Three. Figure Two is typical of the cutter centre line path plots obtained from many of the graphical N.C. tape preparation packages available in the market today; it simply shows cutter movements. Some systems can provide output with lines of different type or in a variety of colours so that one can, for example, distinguish between cutting and non-cutting actions. Even so the information imparted, whilst being of some use to the part-programmer, would mean little to a designer. A technique (McGoldrick and Gibson, 1980) presents the same information in a fundamentally different way; it shows the shape machined in a billet of pre-set size by the cutting actions determined by the tool centre-line path. Non-cutting actions are identified and not displayed. This technique was slow but recent work (McGoldrick et al, 1983) has reduced the computation and display time to a few seconds or less. The outcome is a display of the post-processed data which corresponds to the draughtsman's perspective of the component he intended.

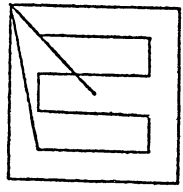


FIGURE TWO

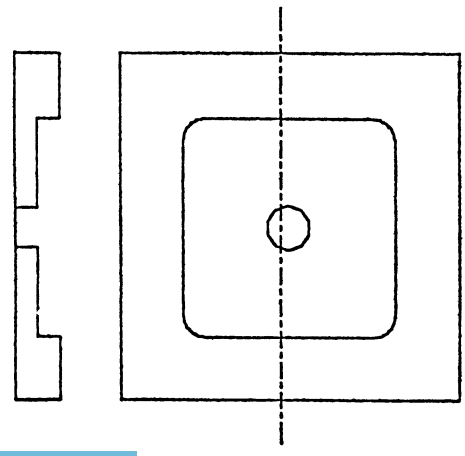


FIGURE THREE

CONCLUDING REMARKS

Attempts to integrate CAD and CAM have not yet started in earnest; modest moves in limited areas are all that have been tried to date.

Integration must be considered from day one; retrofit is not likely to be either possible or successful.

In-house developments are best but costs and skills may be at a premium.

Attention to detail at the specification stage will reap benefits.

REFERENCES

Anonymous (1982) CIM: Total Manufacturing Integration. CAD/CAM Technology, Vol. 1, No. 1, Society of Manufacturing Engineers, Detroit, U.S.A., Spring 1982.

Bonney, M.C., Yong, Y.F., Gleave, J.A., Williamson, J. and Thorne, R. (1983) Developments in the off-line programming of robots using CAD (working title), paper to be offered to Autofact Europe Conference to be held in Geneva, Switzerland, Society of Manufacturing Engineers, Detroit, U.S.A., September 1983.

British Standards Institution (1969) B.S. 4500:1969, Specification for ISO limits and fits. BSI, London, 1969.

Gregory, A. (1982) Weak interfaces in CAD/CAM challenge system developers. Numerical Engineering, Vol. 3, No. 2, Numerical Engineering Society, London, April 1982.

Halevi, G. (1980) The Role of Computers in Manufacturing Processes. John Wiley and Sons, New York, 1980.

McGoldrick, P.F. and Brinsmead, L. (1982) The Economics of N.C. Machining when considering tolerance. Proceedings of the 3rd Polytechnic Symposium on Manufacturing Engineering, Wolverhampton Polytechnic, May 1982.

McGoldrick, P.F. and Gibson, R.K. (1980) A Computer Technique for the Pictorial Representation of the Cutting Action of Three-axis Numerically Controlled Milling Machines. Proceedings of CAD 80. IPC Science and Technology Press, Guildford, April 1980.

McGoldrick, P.F., Lin, Z.-Y., Green, J. and Barson, R.J. (1983) NECTAR - A Graphical Technique to Represent the Component Produced from the Cutter Path of a Numerically Controlled Milling Machine. Paper offered to International CAD Congress

"Data Processing in Design '83", Munich, VDI-Gesellschaft, Dusseldorf, October 1983.

O'Brien, C., Badi, E.A., Güvener, L.Z. (1982) Simulation of turning operations as an aid to process planning. Proceedings of Modelling and Simulation (MS) 82, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (AMSE), Tassin-la-Demi-Lune, France, July 1982.

Parmater, J.Q. (1980) Computer-Aided Engineering: The step beyond CAD/CAM. Machine Design, U.S.A., 23 October 1980.

MODIFIED PARAMETRIC LAGRANGIAN INTERPOLATION FOR CAD

A. Pramila

University of Oulu, Oulu, Finland

INTRODUCTION

In many applications CAD involves both graphical and mathematical modelling of an engineering object. Similarly, it employs frequently mathematical representation of data describing the functional characteristics of the object under consideration. Data may be derived analytically, numerically or they may be based on measurements. Continuous representation of these data as functions of some parameters is needed e.g. for subsequent design calculations or for optimisation algorithms.

An ideal mathematical representation method superior to all others has not yet appeared. Classical polynomial representations (e.g. Lagrangian interpolation) have at least two drawbacks: high degree polynomials implied by a large number of points to be interpolated easily give unwanted oscillations or inflections and slopes of polynomials remain finite for finite abscissa values.

Piecewise parametric splines and patches of Ferguson (1964) type do not suffer from the abovementioned drawbacks but the use of them encounters sometimes difficulties in adjusting the slopes and especially in adjusting the cross derivatives in 3-dimensional cases. Bézier's (1971) splines and B-splines (see e.g. Gordon and Riesenfeld (1974)) have the drawback that the curve does not pass through the points used in defining curve geometry except through the endpoints.

The still limited storage capacity and computational speed of microcomputers do not allow the implementation of e.g. the recent B-spline-algorithms containing the interpolation mode.

The purpose of the present paper is to introduce a new simple parametric interpolation method which seems to have many of the flexible features of parametric splines without their drawbacks. The method has thus far been applied for CAD purposes in Univac 1108 mainframe computer, in HP-87 microcomputer and in Computervision Designer IV CAD/CAM-turnkey system.

FAILING OF THE USUAL LAGRANGIAN INTERPOLATION

In ship design a preliminary lines plan defining the hull form is prepared at an early stage. For subsequent design calculations a curve sheet containing the hydrostatic properties of the hull as function of the draught is needed. These properties may be e.g. displacement (DISP), load required to effect a change in the mean draught at certain waterline of 1 cm (tonnes per centimetre, TPC), distance from keel to center of buoyancy (KB), distance from center of buoyancy to transverse (BMT) and longitudinal (BML) metacentre.

The values of the abovementioned entities are calculated from the lines drawing for some waterlines and are interpolated in a suitable way for the intermediate draughts corresponding to various loading conditions. Figure 1 shows an example of a curve sheet for a ship with length 120 m, breadth 20 m and draught to construction waterline 8 m. Hydrostatic particulars (DISP, KB etc.) have been calculated for draughts 0.8, 1.2, 1.6, 3.2, 4.8, 6.4 and 8 metres. Seven point Lagrangian interpolation polynomials have been employed for obtaining hydrostatic curves as functions of draught. As can be seen BMT and BML curves contain excessive, unnatural bulges between draughts 6 and 8 metres. Thus, usual Lagrangian interpolation is by no means satisfactory here.

USUAL PARAMETRIC LAGRANGIAN INTERPOLATION

Parametric Lagrangian interpolation is in wide use in the applications of the FEM employing isoparametric elements (see Zienkiewicz (1977)). E.g. in two dimensional elements, the parent element is a square in the u, v-parameter-plane (u, v $\in [0,1]$) and it is mapped to the x, y-plane using the same shape functions used in the representation of the unknowns. Thus, in an element with n nodes per side e.g. the side v = 0 is expressed as

$$x = \sum_{i=1}^{n} L_i^n(u)\, x_i \qquad (1)$$

and

$$y = \sum_{i=1}^{n} L_i^n(u)\, y_i , \qquad (2)$$

where x and y are the co-ordinates of the points along the side, x_i and y_i are co-ordinates of the nodes lying on that side and $L_i^n(u)$ are the n-point Lagrangian interpolation polynomials with evenly spaced nodal points in the parameter space.

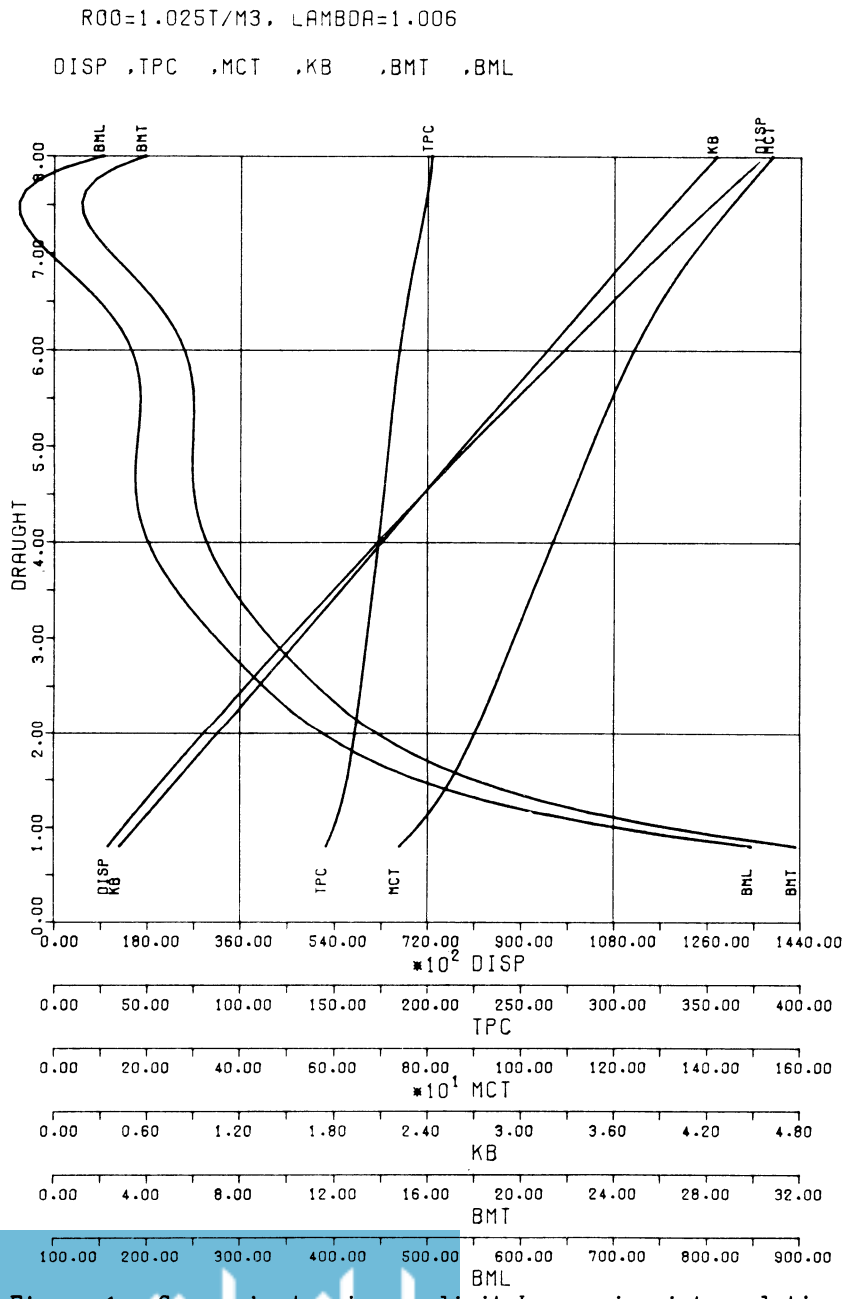


Figure 1. Curve sheet using explicit Lagrangian interpolation.

However, care must be taken in fixing co-ordinates x_i and y_i of the mid-side nodes. If the mid-side nodes are placed improperly, violent distortion occurs. To obtain a reasonable result the well known "mid-third" rule is applied e.g. for elements with 3 nodes per side. Similar rules can be stated for higher order elements for the purpose of smooth interpolation. Generally, the intention of the rules is to ensure that the intermediate nodes are placed evenly enough.

The limiting cases have been carefully studied in some of the applications of the FEM in fracture mechanics (see e.g. Henshell (1975), Barsoum (1976), Pu (1978)). For example, when equation (1) is used to map the parent element on x-axis between 0 and a, the mid-node must not be placed closer than a/4 to the end-nodes in order to avoid mapping outside the interval [0, a].

For later comparisons, an example is presented in figure 2, where there are five interpolation results with same endpoints but with different placement of the mid-point. As can be seen, when the mid-point is too close to the endpoint the interpolation result is unsatisfactory.

In interpolation applications involved in CAD one cannot choose the placement of the points from the interpolation point of view. Thus, the usual version of parametric Lagrangian interpolation does not offer any drastic improvement compared with explicit Lagrangian interpolation.

MODIFIED PARAMETRIC LAGRANGIAN INTERPOLATION

Introductory example

Let us consider the mapping of the 3-noded parent element on the x-axis with nodes $x = 0$, $x = \alpha a$ and $x = a$. Let us further choose - instead of evenly spaced u values -parameter values 0, β and 1 for the knot points used in defining the Lagrangian polynomials $L_i^3(u)$. Now, the shaded region in figure 3 shows the limits between which α may vary for every β to obtain a smooth mapping. Thus, e.g. if we take $\alpha = \beta$ we are always on the allowed region.

Generalisation

From the rather trivial example above and from the previous discussion on the usual parametric Lagrangian interpolation we may deduce at least qualitatively that the smooth result requires that the nodes on the parameter axis and along the interpolation curve must be placed correspondingly, i.e. if e.g. the points to be interpolated are closer to each other at the beginning of the curve then the corresponding points on the parameter axis used in constructing the Lagrangian polynomials must also be closer to each other with small values of the parameter.

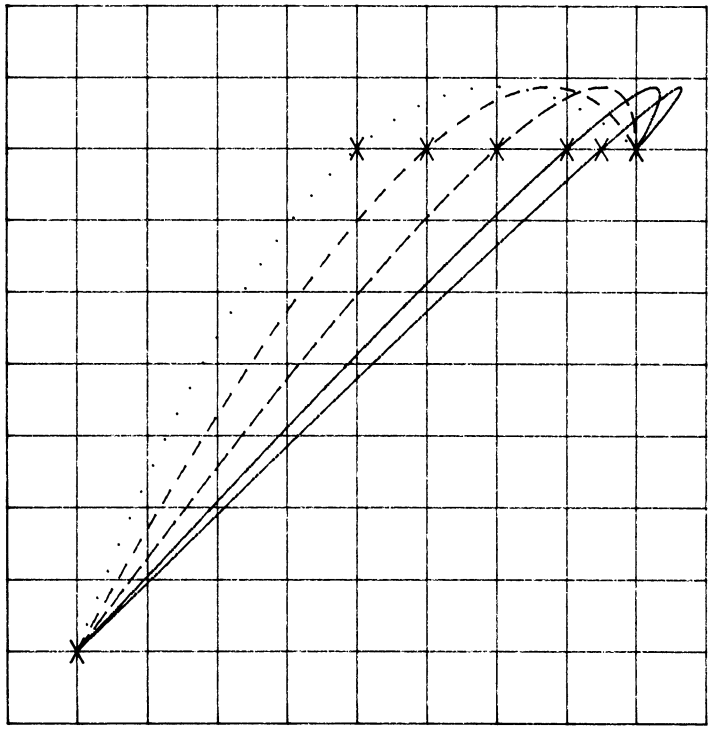


Figure 2. Usual parametric Lagrangian interpolation results.

This is achieved here as follows: Let the points to be interpolated be (x_i, y_i), $(i = 1, \ldots, n)$. The length of the broken line connecting these points is

$$S = \sum_{j=2}^{n} S_j , \tag{3}$$

where the distance between points i - 1 and i is

$$S_j = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \quad (i = 2, \ldots, n). \tag{4}$$

The parameter values for points 1 and n are taken to be 0 and 1, respectively. For other points, the parameter values are chosen according to formula

$$u_i = \sum_{j=2}^{i} s_j/S, \quad (i = 2, \ldots, n-1). \tag{5}$$

Modified parametric Lagrangian polynomials of order n - 1, i.e. for n points, are now obtainable from the well known product

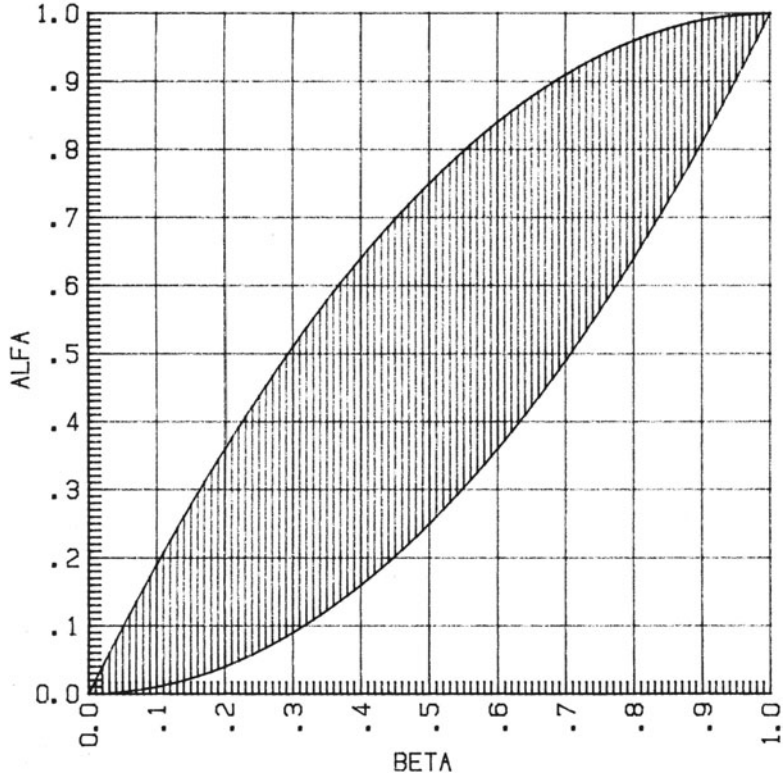


Figure 3. Allowable range of α as function of β.

$$L_i^n(u) = \prod_{\substack{j=2 \\ j\neq i}}^{n} \frac{u - u_j}{u_i - u_j}, \quad (i = 1, \ldots, n). \tag{6}$$

Interpolation is then performed using these polynomials according to equations (1) and (2).

Figure 4 shows interpolation results obtained using the present, modified parametric Lagrangian interpolation with same interpolation points as in figure 2. It can be seen that using the present method the interpolation result remains acceptable irrespective of the location of the mid-point.

To show that the results remain satisfactory - no excessive bulges - even with higher order polynomials, we refer to figure 5, where the same interpolation points (7 points, 6th degree polynomial w.r.t parameter u) are used as in figure 1.

Surface interpolation

Space curves are defined similarly, i.e. the parameter value

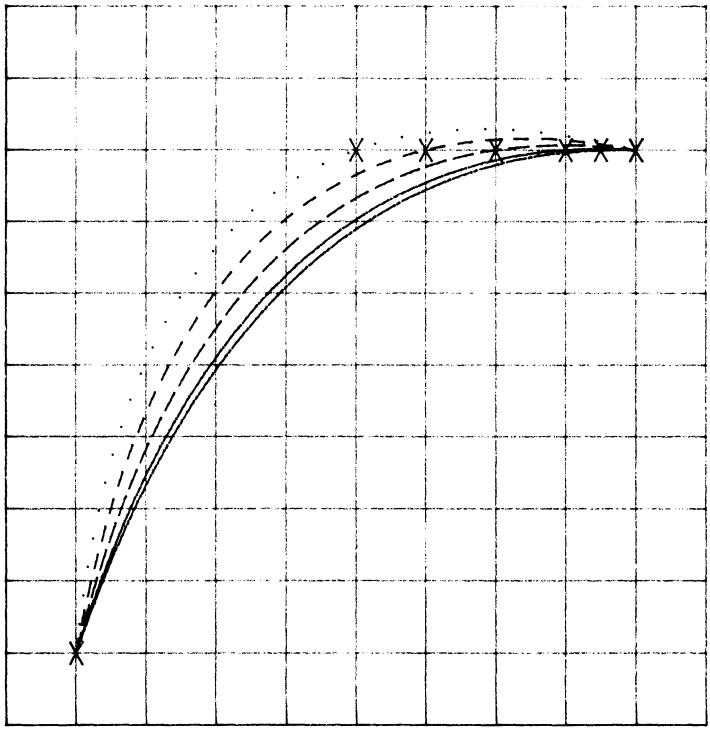


Figure 4. Modified parametric Lagrangian interpolation results.

associated to a point to be interpolated is the ratio of the length of the broken line from first point to the point under consideration to the length of the entire broken line from the first to the last point.

For a surface, a set of curves described by the position vectors $\bar{r}_k = \bar{r}_k(u)$ $(k = 1, \ldots, K)$ are first defined. Then the surface $r(u, v)$ is obtained by interpolating in v-direction

$$\bar{r}(u, v) = \sum_{k=1}^{K} L_k^K(v)\, \bar{r}_k(u). \qquad (7)$$

This kind of surface interpolation causes no restrictions with respect to the number of points used in defining each curve.

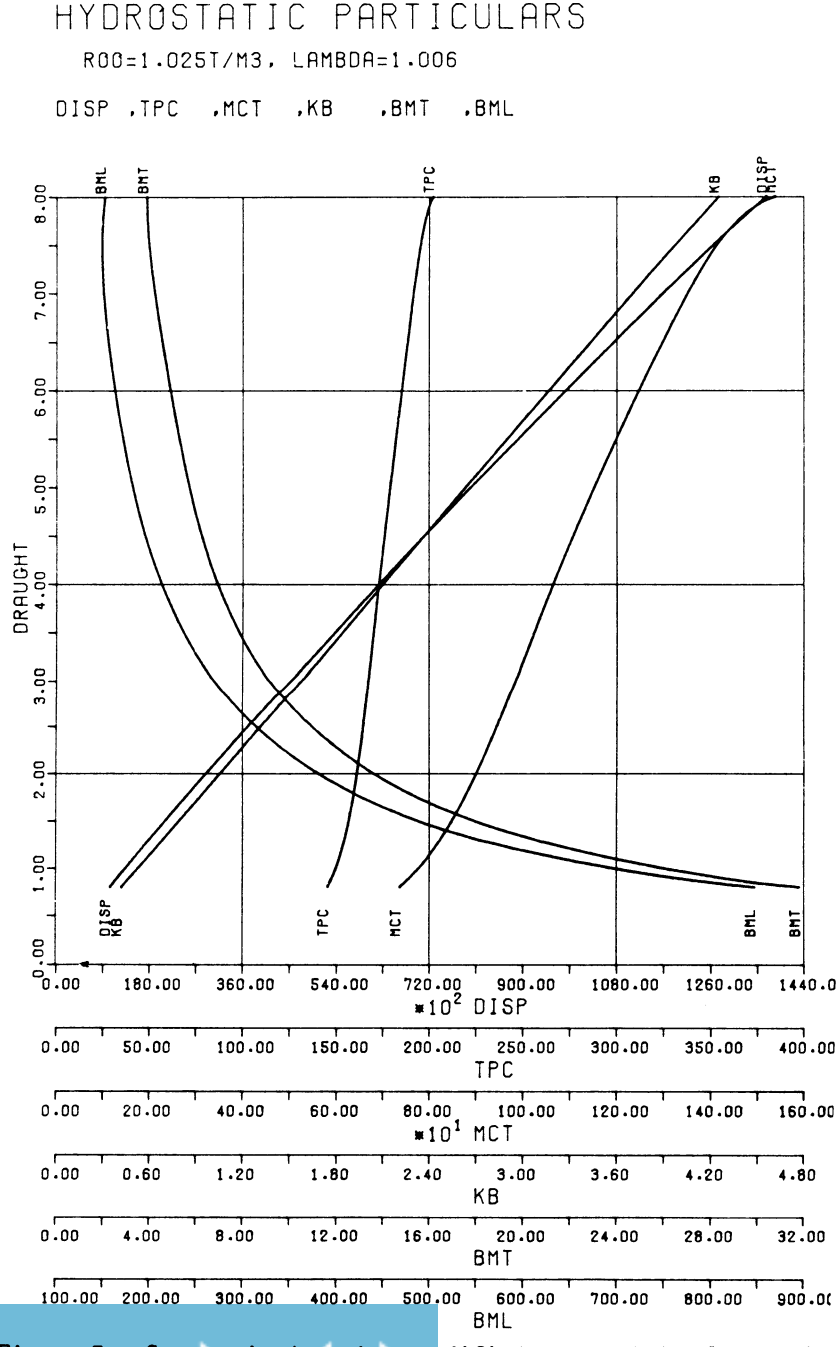


Figure 5. Curve sheet using modified parametric Lagrangia interpolation.

IMPLEMENTATION INTO CAD/CAM-TURNKEY SYSTEM

The Computervision Designer IV CAD/CAM-turnkey system involves in its CADDS3 software system rather wide variety of application programs for graphics handling. However, in surface interpolation with B-spline-surfaces difficulties are sometimes encountered when the number of points used in defining each curve used in creating the surface varies greatly from curve to curve.

The commands of CADDS3 are composed of easy-to-learn English-like words or phrases. Commands usually consist of four parts: verb, noun, modifier and data input, e.g. INS LIN HOR DIG: dd, which inserts a horisontal line beginning from the first point digitized and the length of the line determined by the x - co-ordinate of the second point digitized.

The present algorithms have been implemented into the system in a similar way. Data concerning the points to be interpolated can be entered either by digitizing or by giving co-ordinates. For surface design or interpolation a user oriented input possibility is offered. The curves used in defining a surface may generally be space curves plane projections of which are obtainable e.g. of an engineering drawing. The curves may be inputted by digitizing two plane projections of the curves.

The present algorithms do not restrict the number of points used in defining each curve used in surface creation. Some of the curves can be degenerated to points without difficulties (see Fig. 6).

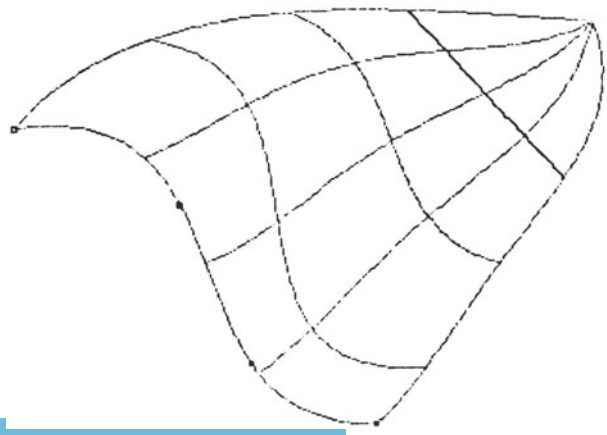


Figure 6. Surface created using modified parametric Lagrangian interpolation.

CONCLUDING REMARKS

A modified parametric Lagrangian interpolation method is introduced for CAD purposes. The fundamental idea of the method is the adaptive manner in which the parameter values used in defining the interpolation polynomials are associated to the points to be interpolated.

The method has thus far bee applied for CAD purposes in Univac 1108 mainframe computer, in HP-87 microcomputer and in Computervision Designer IV CAD/CAM-system. The experiece obtained suggests that the present method has many of the flexible features of more complicated methods without their drawbacks.

ACKNOWLEDGEMENTS

The contribution of Mr. M. Paavola in converting the algorith for the Computervision's CAD/CAM-system is gratefully acknowledged.

REFERENCES

Barsoum (1976) On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics. Int. J. Num. Meth. Engn, 10, 1: 25-37.

Bézier, P. (1971) Example of an Existing System in the Motor Industry: The UNISURF System. Proc. Roy. Soc. Lond., A321, 207-218.

Ferguson, J. C. (1964) Multivariate Curve Interpolation. Journal ACM, 11, 2: 221-228.

Gordon, W. J. and Riesenfeld, R. F. (1974) B-spline Curves and Surfaces. In R. E. Barnhill and R. F. Riesenfeld (eds.) Computer Aided Geometric Design, Academic Press.

Henshell, R. D. and Shaw, K. G. (1975) Crack Tip Finite Elements an Unnecessary. Int. J. Num. Meth. Engn., 9, 3: 495-507.

Pu, S. L., Hussain, M. A. and Lorensen, W. E. (1978). The Collapsed Cubic Isoparametric Element as Singular Element for Crack Problems. Int. J. Num. Meth. Engn., 12, 11: 1727-1742.

Zienkiewicz, O. C. (1977) The Finite Element Method. 3rd edition. McGraw-Hill.

LATEST ADVANCES IN COMPUTER AIDED DESIGN FOR THE PROCESS INDUSTRIES

Vernon T. Taylor

Financial Director
ISOPIPE Ltd.

REQUIREMENTS

The requirements of the process industry are simple, for owners to obtain a well designed and profitable plant at minimum cost and to time, and for contractors to obtain a defined specification not subject to change: both the construction and operation are subject to planning and safety regulations. This ideal is seldom attained, and computer sciences can assist in both design and management activities.

Design Activities

History The traditional method of all disciplines to transmit the design decisions and instructions has for many decades been the pencil and paper drawing, with plan, elevations and sections, with general arrangements cluttered with permanent and ephemeral dimensions. Twenty years ago the use of three dimensional models with copper wire and washers became common, and ten years ago computers to assist in the designers' work began to contribute to the design and construction process.

Drafting and Design Many figures showing the areas in which computer aided drafting and design are now in use, have been published. Figure 1 is typical for plant and pipework, similar figures may be drawn for civil, electrical and control disciplines, but all proceed from information on a process flowsheet, and end at material and erection information. The transition from non dimensional concepts of the flowsheet, and P. & I. diagram to dimensional plant requires that information from the former is expanded, and not lost in the transition.

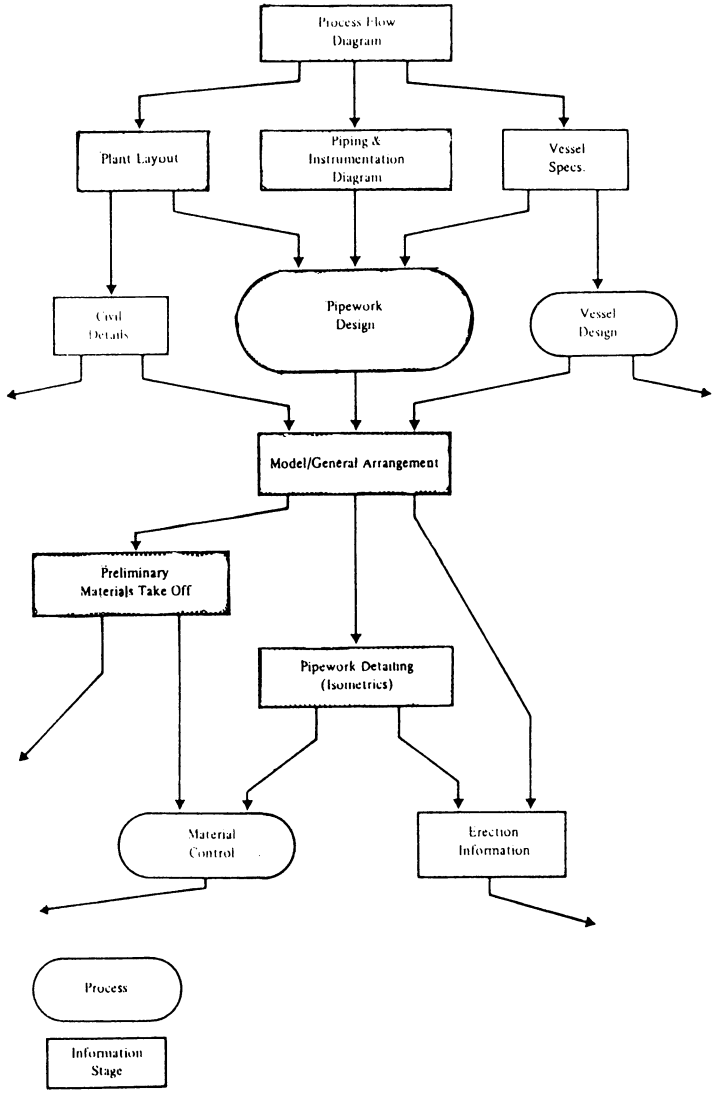
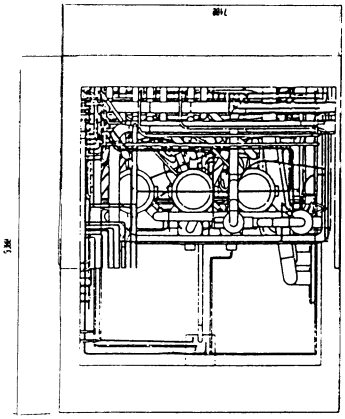


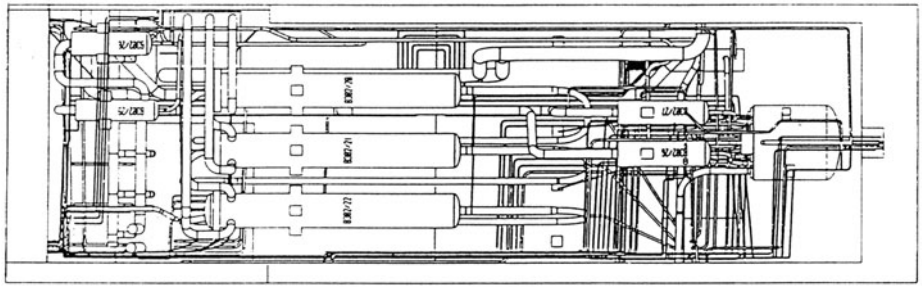
Figure 1 Areas of Computer Application

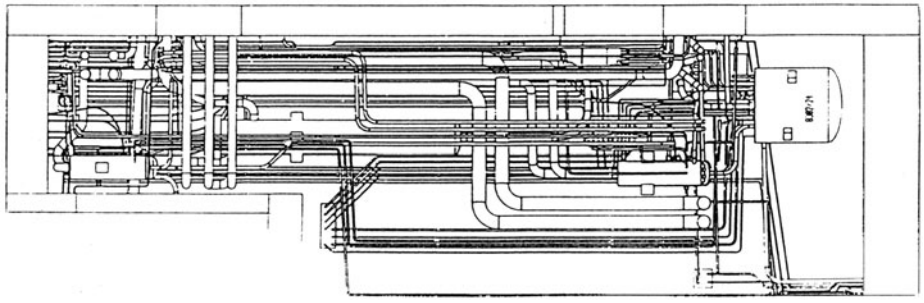
There are many proven two dimensional drafting systems, amongst them, Autotrol, Calma Pegs, Computervision, each with its own merits and limitations, but there is at present only a single proven system for 3D, P.D.M.S. (Plant Design Management System) Crown Copyright, developed at Cambridge by the British Department of Industry's Computer Aided Design Centre, Akzo,b.v. of Arnhem, Netherlands and Isopipe Limited of Nottingham. This programme used for the design of process plants now constructed and working since 1979, has the ability to deal with the plant containing plant items and pipes using the control of multiple and related databases, clash detection and space reservation, and storing spatial data for use in other proprietary isometric programmes, such as Isopaedec, Compaid and Raet.

Figure 2 illustrates a typical view of a dense piping structure from which all material and details for construction offsite are available.

Figure 7 Dense Piping







Design Tools The first distinction to be evaluated is that from turnkey vendors, or from independent sources. The former are restricted by structure to the interfaces available, and the ability and cost of data transfer. CADDS 4 from Computervision has some restrictions, whilst P.D.M.S. developed on Prime Computer is now available in I.B.M. and VAX versions. Whatever programme is selected, the operating terminal, and the interactive response time can offer live screen pictures from which hard copy units for local study are available. The lead established by Tektronix versions is being challenged by other makers, for example PERQ and the advanced workstation from Xtrak Corporation.

Interfaces All stress packages require the details of pipe routing, and a check is necessary whether design packages have the data in their database in a transferable format, or whether individual entry data must be made. Many stress packages are well proven, including P.S.A.5, Triflex, Pipeline, Dynaflex, and are available as bureau users independent of their design methods. Interfaces to isometric and material control packages are available to in-house owners of their own packages from P.D.M.S. and other programmes.

Plastic models The use of plastic models is widespread. The advantages of a model that can be viewed by eye are unique and valuable for planners and for training. The disadvantages rest with the cost of additional models, the assurance that clash detection is valid, and the precise status in the design routine. At least a plastic model is a visual copy of data available in a database, and it may be used as a design tool from which data must be transferred for all other uses such as isometric detailing. It is likely that a consensus of the use of models within and without the computer will be reached, and this was the theme of the American Engineering Model Society 1981 meeting.

Design Management activities
The use of computers to assist in the design process is of limited value, if the extension into the management control and reporting function is not available. The 3D levels of management are shown in Figure 3, Decision at the top of a pyramid, Definition of that decision in the middle level, and Distribution of the decisions to those who require them.

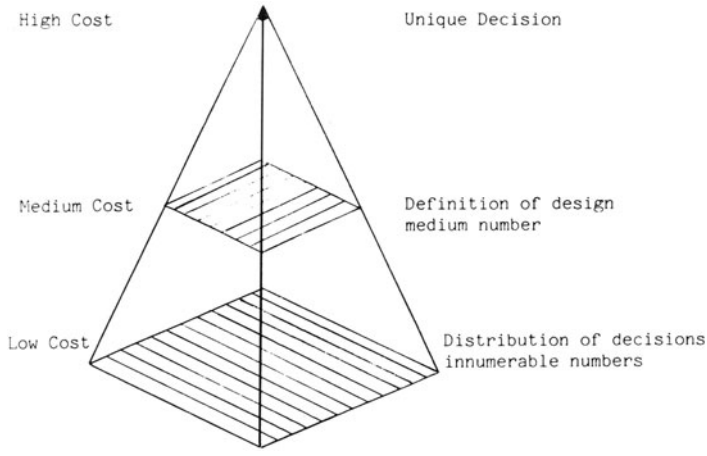


Figure 3 The Decision Pyramid

All CAD systems have some form of reporting required to blend with the contractors purchasing system, progress reports, and finally the as-built data, as distinct from the as-designed data allowing for site alterations. Safety regulations in various locations, especially for hazardous plants, require the owner to have detailed knowledge available at all times, of pipelines and pressure systems containing hazardous fluids and of the precautions necessary to be taken in emergency.

ADVANCES IN COMPUTER AIDED DESIGN

At the time of writing, November, 1982, it is possible to look forward to the advances in the near future.

Terminals The advance in the use of colour and improved response and definition is becoming available. High resolutions (1024 x 1024), vector writing speeds up to 800K, multi pen plotters, raster refresh screens, 300m and pen options in various combinations are in preparation and the choice for the process industries' designer may be to avoid the cost of technical benefits that are attractive but not necessary. Middle range terminals will continue to be adequate for many duties.

Software Additional modules to existing software such as P.D.M.S. are being tested. Rumours of the proof of other wholly 3D systems from Computervision, Genesys, Intergraph and others await a reference of plant design use. The separation of software from turnkey offers will dominate the next few years, since database organisation will require the adoption of processing for the same plant data on whichever hardware is most suitable, or available, from a Cray to the microcomputer.

Hardware The third party to assisted design, hardware, dominated CAD ten years ago, it has now been relegated to third place: the management of databases and the organisation of information predominating.

Conclusion CAD for process plant design is certain to replace older methods. Many futures are being sold, and the careful selection and regular advance is necessary. On presentation in April, 1983 the options will be listed.

REFERENCES

Madden, J. (1980) How to Manage C.A.D.
Institution of Mechanical Engineers C414/80

Madden, J. and Taylor, V.T. (1979) Computer Aided Design,
London. 11, 3: 142-145

American Engineering Model Society (1981)
St. Louis, Missouri Meeting., P.O. 2066, Aiken, S.C.29801.
U.S.A.

Hazardous Installations (notifications and survey)
Regulations 1978. Health & Safety Commission.
ISBN 0 11 883205 0

DEVELOPMENT ASPECTS OF A SMALL C.A.D. SYSTEM

Z.Grodzki and M.Winiarski

Technical University of Warsaw

INTRODUCTION

SNOB is a small computer system, developed for the analysis and partial design of buildings supported on load-bearing walls, usually cross-walls. It grew under the pressure of client requirements-from a short statical analysis program into a system that includes the majority of strength checks required in the calculations by the Building Code. After several years of development and two implementations, the latest additions include new algoritm and program for the calculation of the width of the footings as well as the check on foundation subsidence, introduced recently in the Code for the Design of the Foundations of Buildings

METHODS OF ANALYSIS

There are three groups of methods available for the analysis of interconnected shear walls: Finite Element Method, Wide Column Frame Method and Continuous Medium Method /see Fig 1/.
Their properties can be arranged in a Table /see Table 1/
The Finite Element Method is uneconomic for walls with regular rows of openings, but-as it is the only of the three methods to indicate local stresses-it is used in some cases, when the openings are large and irregular.
Plane Frame programs, available on most computers, usually require serious modification when applied to shear wall problems. First, to allow for the large dimensions of nodes, fully rigid ends must be introduced in lintels. The length of these rigid parts should take into account the elastic encaster-

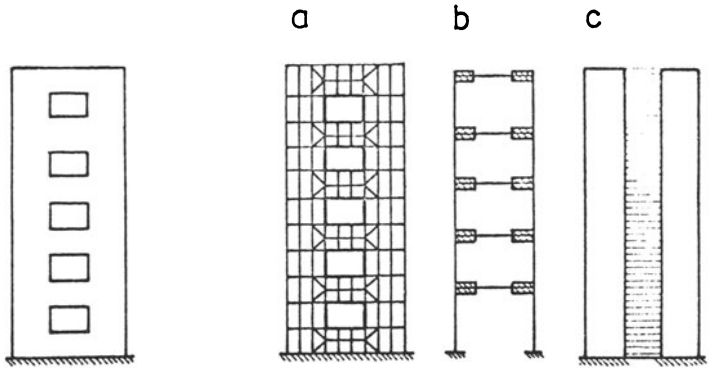


Figure 1 Three methods for the statical analysis of shear walls

TABLE 1

METHOD	VOLUME OF DATA	COMPUTING TIME	ORIENTED TOWARDS	LOCAL ANALYSIS CAPABILITY	FIELD OF APPLICATION
FINITE ELEMENTS	Large	Long	Analysis	Yes	Walls with irregular openings
WIDE COLUMN FRAME	Medium	Fairly Short	Both A & D	Limited	Walls with fairly regular openings
CONTINUOUS MEDIUM	Small	Very short	Design	No	Walls with regular rows of openings

ing of the lintel in the wall. For some ratios of wall width to lintel depth /close to unity/ this length is difficult to establish, as smallest alterations result in large variations in forces and deflections.

With the slenderness of walls and lintels well outside the assumptions of the frame theory, shear deformations cannot be neglected; otherwise, errors of more than 10 per cent can be expected

When, in a frame program at hand, shear deformations are neglected, this can be remedied by multiplying the data input concerning the Moments of Iner-

tia of the relevant rectangular members by a reducing factor

$$K = \frac{1}{1 + 2(1 + \nu)\left(\frac{D}{L}\right)^2}$$

where: ν = Poisson's ratio
D = member depth
L = member length

STATICS PART OF SNOB

The Continuous Medium Method program used in SNOB is simple and displays the properties shown in Table 1.
Although it may be viewed as primitive from a more analysis-oriented point of view, the Continuous Medium Method is very useful in the design of plane interconnected shear walls and their systems.
In the case of blocks of flats of cross-wall construction, the question of torque and rotation does rarely arise in practice, because the spacing of walls along a building /or a part of building between two expansion joints/ is fairly even. In cases when the transverse displacements of all walls can be considered to be equal, the analysis of an entire building can be carried out at one go, by joining the parallel interconnected walls via imaginary rows of very flexible connections, thus assuring their equal deflections while not transmitting any shear along these rows /see Fig. 2/.
This approach was found quite satisfactory for the design purposes in most cases. What it means in terms of computer demands in the case of a 15 storey, 20-wall building when applying the same approach to plane frames or plane plates using F.E.M., can be seen from Table 2.

TABLE 2 Number of unknowns /15-storey, 20-wall building/

METHOD	FINITE ELEMENTS	WIDE COLUMN FRAME	CONTINUOUS MEDIUM
Nº OF UNKNOWNS	1400-4000*	560**	19

* depending on the type of elements
** taking into account extra nodes required by the Wide Column Method

The volume of data required is in proportion to the number of unknowns, unless efficient data generators are available in F.E.M. and frame programs.

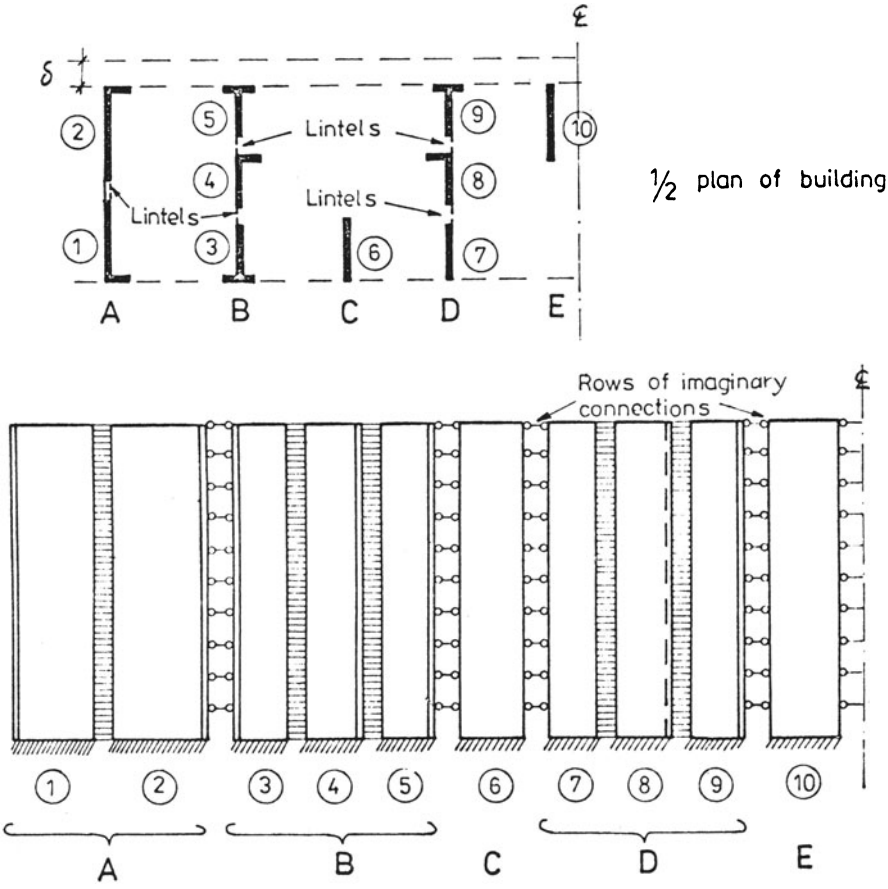


Figure 2. Method to deal with parallel wall systems

The computing time is also in proportion to this number, assuming equal bandwidth in all cases.
It must be stressed, that this approach is unsuitable in the case of skeletal buildings, with a few shear walls or cores placed unsymmetrically in plan to that the centre of rotation of the building is situated far from the line of action of the resultant wind force.
The solution of the problem consists in defining the function T(x), which gives the distribution of the unit shear along a row of connections This function is obtained by differentiating the energy equation, reducing it to a set of algebraic equations and finding the unknown parameters of this set. The approximate method of solution /Rosman, 1964/ assu-

mes a constant distribution of unit shear over the entire height of a row of connections, whereas in reality this distribution is pear-shaped /see Fig. 3/. The maximum shear in lintels occurs at about 1/3 of the height and can be derived from the average shear by multiplying it by a coefficient ranging from 1.4 to 1.9, depending on the stiffness of the lintels.

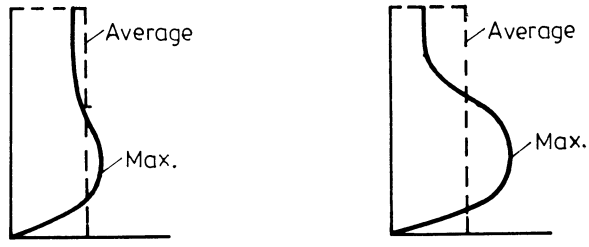


Figure 3 Unit shear distribution in a row of connections

The statics program of SNOB contains some refinements:

- elastic encastering of lintels in the walls has been taken into account
- shear deformation effect has been allowed for in the case of rectangular and T - shaped lintels, but ignored in the case of connections by floor slabs only.

DESIGN PART OF SNOB

Starting with the simple statics program discussed above, the development took place in the direction of strength calculations and checks required by the Building Code. The development included:

- subroutines for checking the factor of safety of a wall panel at ground floor and basement levels, taking into account all of the excentricities due to the inaccuracy of alignment, deviation from the vertical and non-homogeneity of concrete /when the panels are precast in horizontal position/, slenderness and stiffening effect of walls abutting at right angles
- subroutines for checking the factor of safety of horizontal joints at ground and first floor levels

- subroutines for checking the factor of safety of lintels
  - or vertical joints between precast panels-at the point, where the maximum occurs
- subroutines for calculation of the width and the reinforcement of the footings, based on the allowable pressure on the ground

The applications proved particularly successful in the case of system-built blocks of flats, where the input could be reduced to the minimum because of the fixed material properties, known precast element weights and modular dimensions within a construction system.
Within a year of its introduction, SNOB was extended to cover three similar - but differing - large panel construction systems and - in 1980 - a widely different fourth system.

Latest additions

The introduction of a new code for the Foundations of Buildings necessitated further additions and prompted some improvements.
A new method of calculation of the width of the footings was introduced, based on the limit strength theory and was extended to cover arbitrarily stratified ground conditions.
In homogeneous ground, the footing width is computed from a 3rd power equation of the form:

$$K_1 B^3 + K_2 B^2 + K_3 B + K_4 = 0$$

where: B - width of the footing
K_1 to K_4 - coefficients depending on the properties of the soil /self weight γ, cohesion c, angle of internal friction φ, foundation depth and loading

In the case of stratified ground, when the poorest soil stratum is situated at a depth h below the foundation level, this equation should be solved for the soil having the lowest properties and for dimensions of the footing increased due to the spreading of the load, as shown in Fig. 4.
Further requirements of the new code concern the calculation of the coefficient of differential subsidence, which is defined as the maximum relative difference between the subsidence of the three consecutive footings of a building /see Fig. 5/.
The subsidence of each footing consists of two parts, one occuring during the erection and the other during the period of the occupancy of the building.
Each of the parts depends in turn on the load on that particular footing and on the influence of

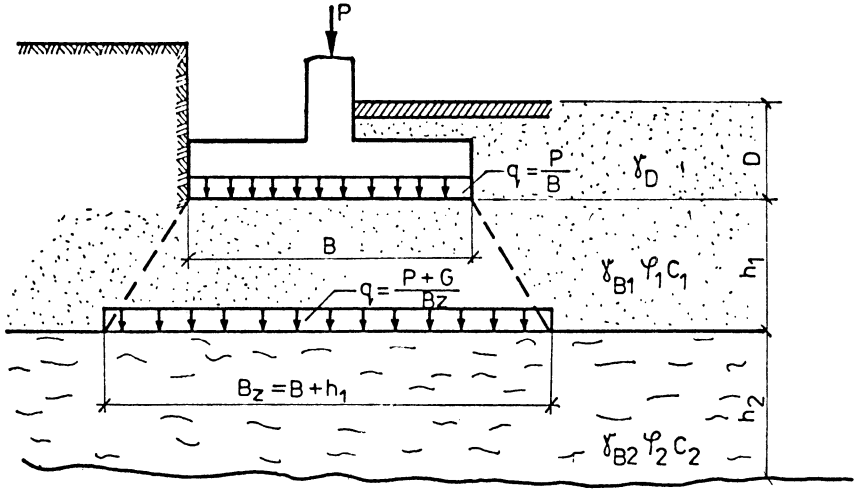


Figure 4 Equivalent footing width in stratified soil

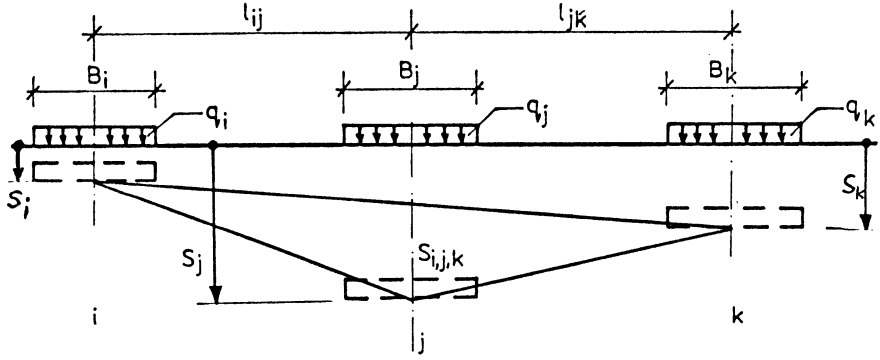


Figure 5 Evaluation of differential subsidence coefficient

other footings

$$S_i = S_{ii} + \sum_{k=1}^{NF} S_{ik}$$

where: S_{ii} - subsidence of the i^{th} footing due to the load on that footing

S_{ik} - subsidence of the i^{th} footing due to the load on k^{th} footing

NF - number of footings

The coefficient S should not exceed 1.0 cm.
Some simplifying assumptions concerning the shape of the footings have been made in the calculation of the subsidence:

- the footings have to be brought to a rectangular shape of equal plan area as shown in Fig.6 The "smoothing" is carried out by a subroutine

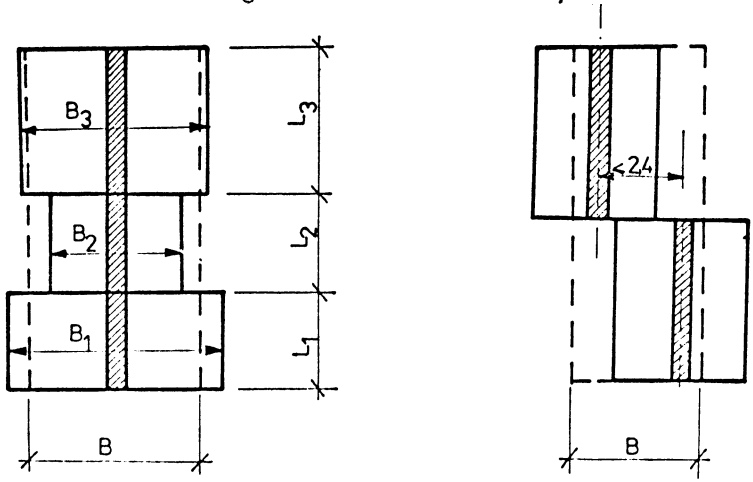


Figure 6 Simplifying assumptions regarding the footings

- the footings are assumed to be rigid and the structure above - of negligible stiffness

The last assumption is justified for buildings supported on cross walls only, with floor slabs spanning between them. The assumption lies on the safe side, because any restraining action of floor slabs tends to diminish the differential subsidence When longitudinal load - bearing walls occur, the designer must intervene and decide on either of the three ways of action:

- lump together the footings and treat them as one rectangular footing of equal area, with the load equal to the sum of the loads
- ignore this group of footing and select the points at which the subsidence is calculated under the remaining footings
- resign altogether from the subsidence computation by switching off this path

The addition of computations concerning the width and the subsidence of the footings doubled the amount of data required by the system, despite the fact, that geometry and load data are derived automatically from the previous data.
Unfortunately, the data concerning the properties

of the soil strata above and below the foundation level i.e. thickness of stratum, unit weight, angle of friction, cohesion, compresibility, etc. cannot be generated in any way The length of computation has also trebled. In these circumstances, the subsidence calculations have been made optional.

SOFTWARE DESIGN

SNOB is at present implemented on ODRA 1300/ICL 1900/ICL 2900 series computers. It consists of about 50 modules written in ANSI FORTRAN IV. No modules are written in autocode /PLAN/ and only the input and output modules use FORTRAN 1900 mainly because of the convenience of the free format.
The modules are short and serve to compute distinct parts of the algorithm that can be treated separately, independently of the system. The flow chart (Fig. 7)underlines this modular character; geometrical properties of separate walls, those of the lintels, properties of interconnected walls, setting up of a set of equations, its solution, various checks, foundation width, reinforcement, subsidence are computed by specialized modules. SNOB grew from about a dozen modules five years ago to the present number by including new modules. This was not carried out automatically, as the authors are of the opinion that organising a comparatively small system round a data base and creating facilities for automatic appending of further modules would increase the "internal administration" of the system and result in longer computing times, especially in the case of smaller problems. The provision of such facilities serves its purpose when the inclusion of new modules is done by the users themselves. The experience of the authors points to the fact that the proposals for extensions, come usually from the users to the organisation responsible for the development of the system or directly to the authors. When a system reaches the size when further extensions become hardly manageable, some remedies are necessary.
The facilities to include new modules can be, however, kept outside the system in the form of suitable aids. A table of input and output parameters was found to be of much assistance /see Table 3/. If an input parameter has its value altered within a module, the output value is given a different name, to avoid any possible future complications. By means of such tables the development of a system is facilitated, especially in the case when several people write separate modules simultaneously. The

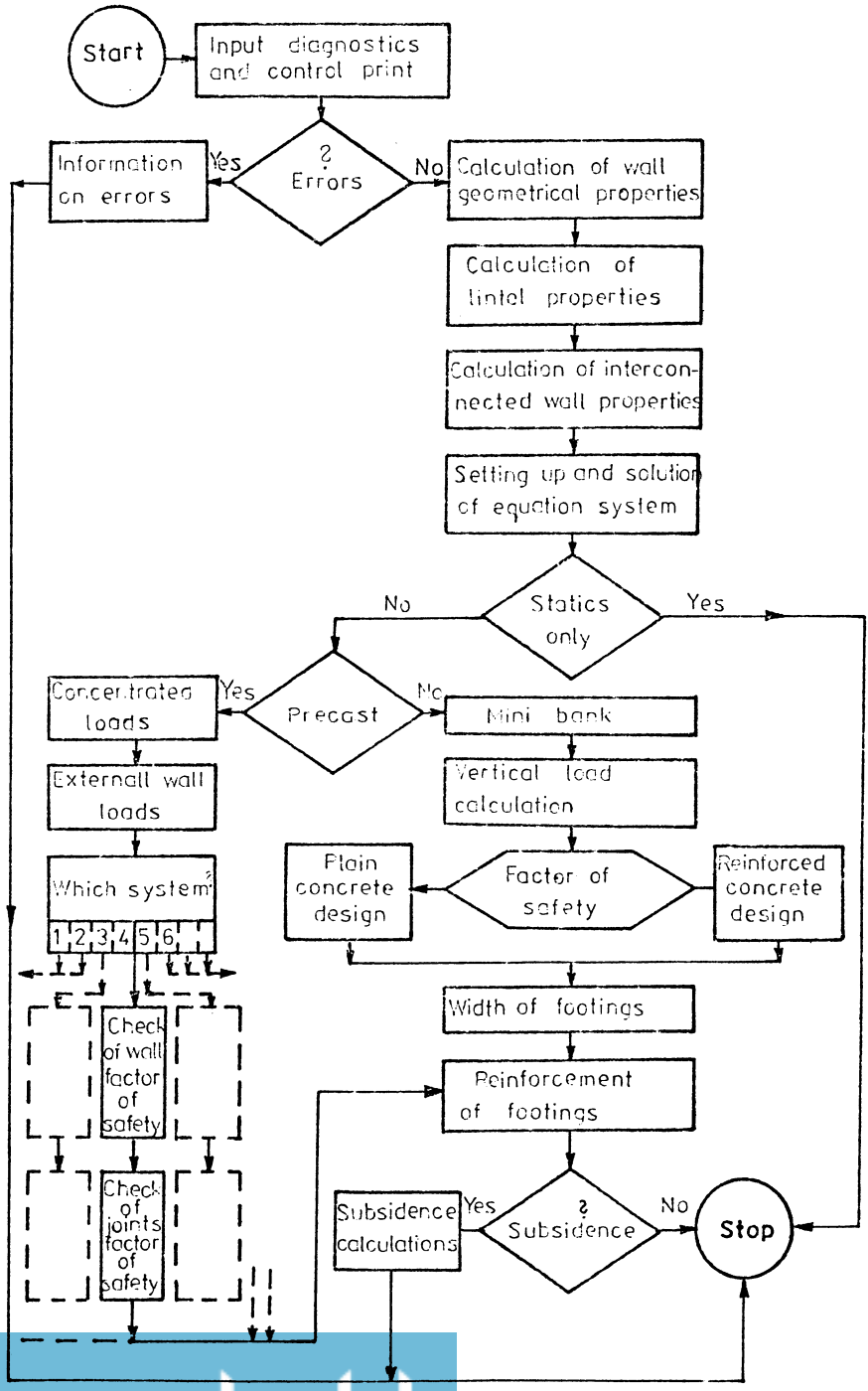


Figure 7 Flow chart of SNOB

Table 3 List of I/O parameters in modules

Parameters / Name of module	NW	NL	NS	H	B	D	TS	A		T	FW
GEOM	I			I							
GAUSS								I		O	
NEWMOD			I				I				O

above table is in such a case a "contact field" of the team The method was tested on a smaller system before applying it in SNOB.

THE DEPENDENCE ON THE BUILDING CODES

In its design part, SNOB has to deal with several standards and detailed calculation instructions for each type of industrilized construction system.
In the superstructure part, SNOB depends on standards for loading in buildings, snow and wind loads, Codes of Practice for reinforced and plain concrete; in its foundation part - with the code for the foundations of buildings.
The detailed instructions for the method of calculation of an industrialized construction system are usually valid for the entire life of that system, that is about ten years.
The standards change at about the same or a faster rate It is an unpleasant feature of the systems which leave pure statics or dynamics field and enter into the design area, that-when they have to deal with several Building Codes - they require an almost continuous stream of changes.
The usual method to deal with these is to organize Standard - dependent parts into separate subroutines. This is, however, not always sufficient when the design philosophy, underlying the Standards is drastically changed
That was the case with the latest introduction of the Code for the foundations of buildings The previous Code required only an input of allowable bearing pressure and a vector of foundation depths below a datum level, whereas the new Code requires vast amounts of data on the properties of each stra-

tum. This caused serious modifications in the input module.

SYSTEM PERFORMANCE

The start-to-stop computing time of just over 1 min for an average building with 30-40 walls increases to 3-5 mins. when the foundation subsidence computations are included, depending on the number of soil strata and points at which the subsidence calculations are required. The relevant C.P.U. times are 10 secs and 1.5 to 2.5 mins. The core area needed is 27 K-words for the master module with one or two overlays.
SNOB operates normally under GEORGE 3 operating system in local or remote batch modes on computers with 64K words core; it has dialogue input available on computers of more than 64K.

CONCLUSIONS

On the basis of several years of experience in development of this system, the main advantages of the approach presented in this paper seem to the authors to be as follows:

- the growth by joining separately tested modules to the existing system that has been thoroughly tested in practice and can be considered substantially error-free, results in minimum number of errors;
- the ease of the management of such a growth is much greater than the introduction of a larger system to users in its final /or nearly final/ shape;
- the economics of such a growth are also more favourable than the development of larger systems /including, perhaps, a problem-oriented language/ and presenting it in a fully developed form:
  first - the approach presented here results in an early commission of a small system, which brings the revenue fairly soon, whereas the development of a system up to its final form freezes the funds for a longer period;
  second - the risk of finding that the completed system may find little favour with the clients is reduced to the minimum by taking the "Organic growth" approach, in which the direction of the development to follow depends - to a great extent - on the changing requirements of the users.

DATA PARAMETRIZATION IN A PRE-PROCESSOR FOR COMPUTER-AIDED DESIGN OF ELECTROMAGNETIC DEVICES

B. Ancelle - D. Boillon - J.L Coulomb

ENSIEG - Laboratoire .d'Electrotechnique - Grenoble (France).

INTRODUCTION

The finite element method is now a standard numerical procedure for solving electric or magnetic fields. Several packages using this method have been built for the design and the computation of electromagnetic devices.
FLUX 2D, developed by the Laboratoire d'Electrotechnique de l'Institut National Polytechnique de Grenoble is an interactive system in which graphic facilities are combined with a convenient command language. [Sabonnadière J.C (COMPUMAG 81)]. The system can be divided in three main independant parts :
- a pre-processor ENTREE for the geometric definition of the model, the finite element cutting out, the allocation of physical properties and boundary conditions [Ancelle B. (COMPUMAG 81)]
- a computation processor RESOL, in which non linear equations occuring from finite elements are solved by a Newton-Raphson or Newton-Kantorowitch method. This program may use either vector or scalar potential and solve axisymetric or cartesian domains in magnetostatic or magnetodynamic hypothesis. Its particularity in regard of others similar programs is to use second order rectangular finite elements which give the best precision in the results of the computation.
- a post-processor EXPLOI, a very interactive and graphical program in which the values of the vector or scalar potential are converted into engineering quantities (field, flux and current density,...) computed in any point of the domain. Moreover, integral quantities like strengths, torques, energies, inductances, capacitances... may be computed with this program. Two main graphical possibilities are offered by the post-processor : the display of equipotential lines in an area choosen by the user : a perspective view for the visualisation of the field strengths at the center of all the elements.

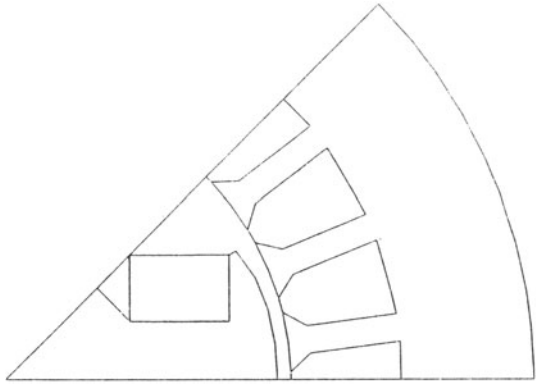


Figure 1a : A.C. Generator

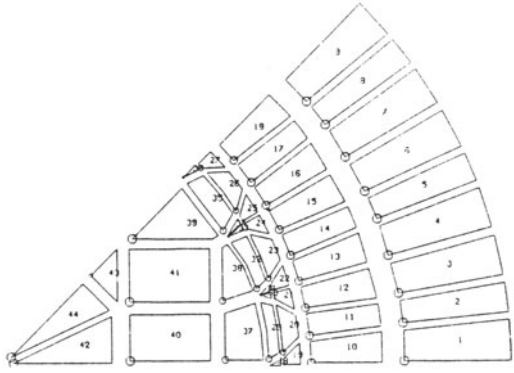


Figure 1b : Finite element cutting out of the device

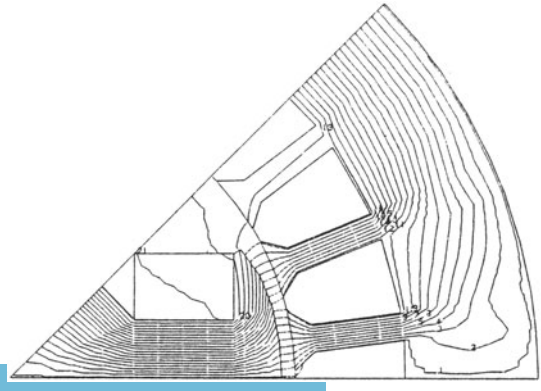


Figure 2 : Example of equipotential plot

FLUX 2D package has been used by several French industries for about two years. Users remarks allowed us to know the main qualities and the main deficiencies of the system. It seems to us that the main difficulty occured with finite elements programs like FLUX 2D is the manipulation of a great number of geometric and physical data in the pre and post-processor. So, a particular effort is in progress to increase the power of the pre-processor and to improve it in the way of a more simple and more interactive use by developing a highly structured data base and advanced graphical CAD techniques. [Ancelle B. (COMPUMAG 78) ; Coulomb J.L. (82)] .

ENTREE : A PRE-PROCESSOR FOR GEOMETRICAL DATA INPUT

The pre-processor is ordered to fulfil three tasks : geometrical definition of the 2D domain, finite element cutting out, allocation of physical properties and boundary conditions. According to these specifications, the pre-processor has been divided in a three models mini data base : geometrical model, discretized model, physical model.
Building the geometrical model is made by using a boundary description. Any device is considered as the gathering of several patterns, each of them made with one precise material. Every pattern is made of several meshes such as quadrangles or triangles. At last, every mesh is made of three or four lines, such as segments, arcs of circle or arcs of parabola; each line, itself, is defined from elementary components such as scalar (lenght, angle, radius, ...) and points coordinates. So the user must first define elementary components and then build, step by step, lines, meshes, patterns, eventually images of patterns by using geometrical transforms (translation, rotation, symetry, ...) and, at last, the final model. Only particles as scalar or points coordinates are numerical items ; all the other components are relational items : by example, a segment is defined from the references of two points. This structure has the main advantage to make numerical modifications very easy to manage.
The discretized model is obtained from the geometrical model by subdividing it into finite elements (rectangle or triangle) according to a successive refinement process. The subdivision is made in an interactive way ; each mesh of the initial domain is cut out by the designer with the help of a menu displayed on the graphic terminal screen ; each new mesh, itself, can be cut out according to the same procedure so that the user can get as many meshing levels as he wants.
The physical model contains three components :
- the physical properties which are reluctivity (mangetostatic) or permittivity(electrostatic).
- the sources which are current densities or electric charges.
- the boundary conditions which are known values assigned to the potential (Dirichlet, Neumann,...).

If we analyse the possibilities of the pre-processor, we can put forward its main qualities :
- it's a very efficient and versatile tool.
- it lies on a very powerful command language.
- it's of a very simple use even for people having no particular knowledge about finite element problems.
- it makes easy every numerical change keeping the topology of the domain constant.

But, we can put forward its main deficiencies, too :
- it's not built for managing a great number of similar models : FLUX software uses a lot of files and obliges the user to define his own files management. By example, a tree structure of models data may be defined with composite file names. A model name is defined for each geometrical pattern. When physical data are assigned, a suffix name is asked to the user and added to the previous name. So each file name is made up from the composite model name and a fixed name.
- it's not enough parametric : one of the main goals of CAD in electromagnetic applications is the optimisation of devices design. Very small geometrical modifications may cause important variations of engineering quantities. With the system described just before, studying the variation of an electromagnetical quantity with a geometrical one obliges the user, for each value of the geometrical parameter, to define the new geometry of the problem and to run all programs.

So, if we really want the system to be an efficient design tool, we must make possible a parametrization of geometrical data and its transmission up to the solving and exploitation processors [Massé Ph. (COMPUMAG 81)].
- the geometrical data input is very simple but may be long, tedious and constraining for big size models. Before data input, the designer must have prepared a dimensioned model of the device made with an assembly of initial and non-discontinuous meshes. Every point must be located by its coordinates, every mesh must be described in trigonometric sense,... For a big model, we can easily reach one or two hundred points, twice more lines,... and the description of the geometry can take several hours. So, if we really want the system to be an efficient design tool, we must improve geometrical data input by using advanced graphical CAD techniques.

INTRODUCING CAD TECHNIQUES AND DATA PARAMETRIZATION FOR A NEW PRE-PROCESSOR

The efficiency of a design tool is tightly linked to the interaction level existing between man and machine. Let us examine the interaction degree introduced in every part of the pre-processor.
- the description of the geometrical model is poor in man-machine dialog. Geometrical data input is made according to a non-directive procedure ; no tool has been developed for helping the user to remember, if it's necessary, the

different commands.
- the description of the discretized model lies on a rather important dialog with the use of a menu for the cutting out of the domain.
- the description of the physical model is highly interactive: by example, when describing the physical characteristic of a material, the user may interactively smooth the B(H) curve on the graphic screen.
So we see that improving the pre-processor means improving geometrical data input. In this way, two new concepts have been introduced in ENTREE :
- use of the Constructive Solid Geometry (C.S.G.) method applied to 2D areas for the description of the model.
- parametrization of geometrical data by using symbolic formulae.

The Constructive Geometry Method
It differs from the boundary representation in that its users design components by working directly with computer representations of solids rather than with points, lines and assemblies. Furthermore, the solids remain valid as the user performs operations on them.
This method is a very general one and can be applied either to 2D or 3D domains. For our system, we developed it on 2D areas ; the user can create any basic primitive surface (triangle, rectangle, circle) directly by specifying its size, location and other parameters. He can combine any two solids with the set operators union, intersection, difference and use any geometrical transform (translation, rotation, symetry,...).
Constructive geometry is a natural and powerful design mode because set operators have analogs in manufacturing (machining, cutting out, ...). But it presents many more advantages:
- the data structure resulting of this description of the geometry is a real tree structure and not a network or plex structure as the one obtained by using boundary representation.
- consequently, it allows an easy transmission of physical or topological properties which have only to be established on basic primitives.
- it guarantees the integrity of all components ... Nevertheless, boundary representation can't be completely left : it's now only a temporary data base used for some specific functions.

Data parametrization
Field computation packages are called upon more and more frequently to be integrated into CAD systems. They must permit their users to adapt interactively the geometry and the physical characteristics of the device in order to satisfy their technical requirements. So it's fundamental to be able to display the effect of the variation of one or more of the device's parameters. In present systems, it's necessary to return to the input level for any geometrical or physical

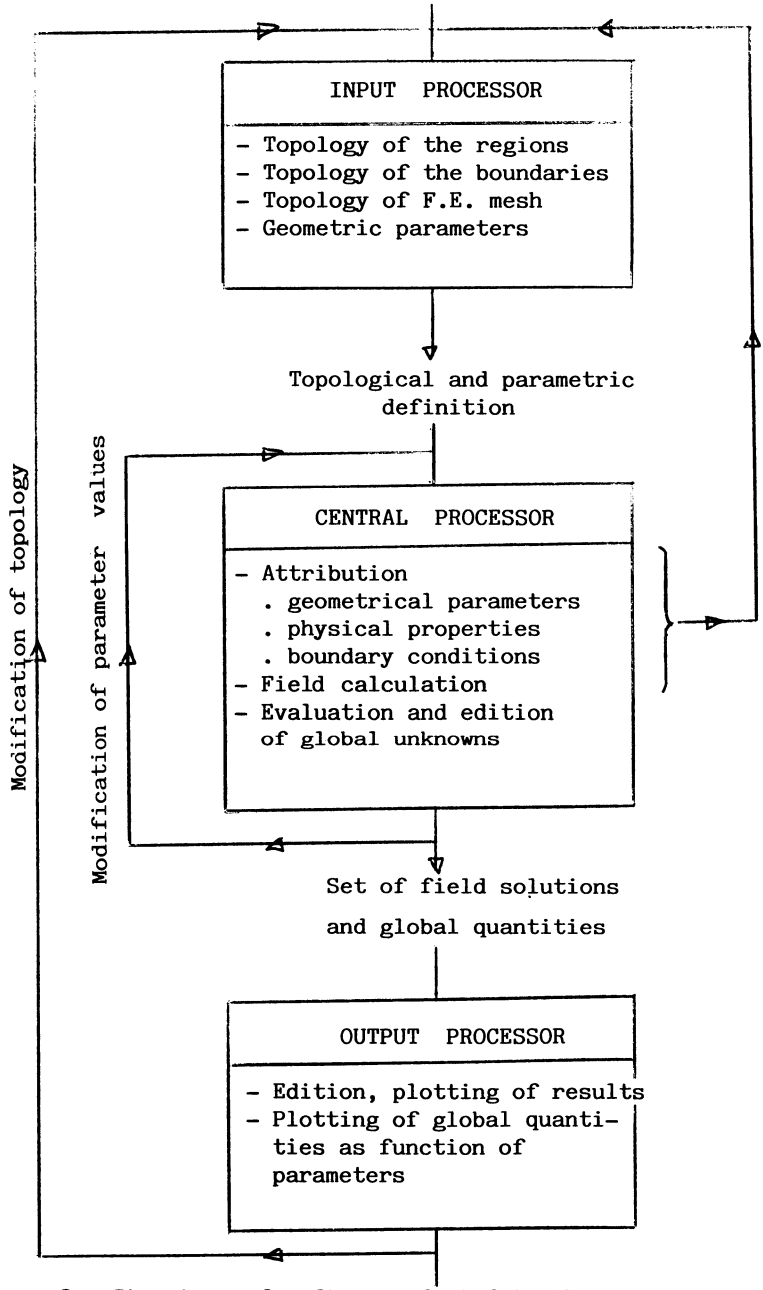


Figure 3 : Structure of software adapted to the parametric definition.

modification. A parametric definition of dimensions and physical characteristics allows to avoid this disadvantage. So we can describe the general structure of software adapted to the parametric definition (Figure 3).
The originality of this approach is to consider any scalar entity or point coordinate as a parameter and not as a fixed value. Geometrical parameters are managed in a bank and transmitted up to the computation and exploitation processors.
But what's a parameter ? A parameter is an arithmetical expression to which a symbolic name is affected in the input processor. A parameter can be function of other parameters : we speak then of formula. So we can define a formula as an arithmetical expression made of real numbers, arithmetical operators (+, -, *,/,**,...), usual functions (SIN, COS, LOG, ...) and different parameters. The main possibilities given to the user for parameters manipulation are :
- to create interactively or by program a formula (or parameter)
- to affect a numerical value to a formula (or parameter)
- to modify or break up a formula (or parameter). But the modification or destruction of one of them may break coherence of the data structure using it because parameters and formulae are organized in a network structure. So management tools have been created for preventing from an inopportune modification.
- to access to the bank for having knowledge of the present numerical value of a parameter (or formula) or the text of its arithmetical expression.
- ...

Data parametrization in the input processor improves largely CAD systems : it allows the user to build a class of models (only different by their geometry or physical characteristics) in using only one command, it avoids to store input orders and to execute them again in batch processing for a geometrical modification of the device. Then the loop for the design process is reduced to a minimum because it lies in the central processor.

CONCLUSION

The research in electrical engineering and the development of software tools to compute electric or magnetic fields has lead to the FLUX system which implements a lot of CAD techniques. The new improvements brought to the pre-processor by using the Constructive Geometry Method and the parametrization of the geometry and physical characteristics of a device made us build a very interactive and fully parametric input processor. It avoids the tedious manipulations of a lot of geometrical data which was restraining the developement of F.E. programs in an industrial environment. We can now consider that FLUX 2D is a very efficient and versatile tool for the design of electromagnetic devices.

REFERENCES

Ancelle, B. (COMPUMAG 78) Definition and drawing of electrotechnical devices for field calculation. Grenoble.

Ancelle, B. , Callegher, E. , Massé, Ph. (COMPUMAG 81) : ENTREE : a fully parametric pre-processor for computer aided design of magnetic devices.
IEEE Transactions on Magnetics, MAG 18, n° 2, March 1982.

Boyse, J.W. , Gilchrist, J.E (March 1982) GM Solid : Interactive modeling for design and analysis of solids.
Computing Graphics and Applications.

Coulomb, J.L (1982) A simple interactive finite element processor for teaching purposes.
Journal of Applied Physics.

Massé, Ph. , Coulomb, J.L , Ancelle, B. (COMPUMAG 81) System design methodology in CAD programs based on the finite element method. (Invited paper).
IEEE Transactions on Magnetics, MAG 18, n° 2, March 1982.

Requicha, A.A.G (December 1980) Representations for rigid solids : theory, methods and systems. Computing Surveys, Vol. 12, n° 4.

Sabonnadière, J.C , Meunier, G. , Morel, B. (COMPUMAG 81) FLUX : A general interactive finite elements package for 2D electromagnetic fields.
IEEE Transactions on Magnetics, MAG 18, n° 2, March 1982.

EXPERIENCE WITH COMPUTER AIDS FOR ROUTING PIPES - ISOROUTE

A G Foord

Design Systems Group
ICI Engineering Department

SUMMARY

Nine different ICI design projects for chemical plant have used an ICI program called ISOROUTE or the CAD Centre program PIPER [Newall 1972] to assist designers. A review of these projects reveals the value of a computer program which can quickly select routes for pipework which avoid obstructions. Both project and pipework designers reject "automatic routing" for pipework. Project designers will, however, use ISOROUTE as a tool to assist their own design.

INTRODUCTION

A project or pipework designer or engineer frequently starts work with incomplete, inaccurate, and inadequate data. Typically: (a) preliminary schematic process flow sheets; (b) one or more possible sites for the project (usually sites of different shapes); (c) vague indications of the services needed and (d) an unrealistic time-scale.

Computer programs for CAD, by contrast, usually need:

(a) accurate information;
(b) complete sets of data;
(c) updating of specifications; and catalogues; and
(d) objectives that do not conflict.

This type of information can be provided only by waiting until the later stages of the project program as illustrated in Figure 1. Use of the program PDMS linked to ISOPEDAC has saved design time and eliminated some design errors in these later stages [Allen 1981, Foord and Rowe 1981]. Recently, however, it has been possible to make an impact on projects at a much earlier stage by

using a computer-aided pipe routing program called ISOROUTE with PDMS. ISOROUTE will find routes for pipework around obstructions and along pipe racks, even when operating with simplified or incomplete data.

CRITERIA FOR SUCCESS

There are three significant measures of success in the design function:

1) design labour (direct cost)
2) design quality (indirect cost)
3) elapsed time.

Most computer aids concentrate on the later and more labour intensive stages of design to reduce effort and minimise mistakes. ISOROUTE and PIPER attempt to aid the earlier stages and could in principle be used in several of the stages.

Example Problem - Project A

Consider a chemical plant with 100 items of equipment, 300 pipelines, 4 floors of multi-bay steelwork with cable trays, stairways, lifts, access paths and a variety of other obstructions.

How do you define:

(a) the "best" pipe routes?
(b) the "best" estimates for some 300 pipes at an early stage in the design process?

Even having defined what the "best" means it is still a difficult problem to define several hundred obstructions and calculate 300 best routes. This particular example was tackled using ISOROUTE. It took 3 man-weeks to assemble and enter all the data into a computer, and only a few minutes to calculate the routes shown in Figure 2.

AVAILABLE SOLUTIONS

Mitsui Engineering and Shipbuilding, Niigata Engineering, Toyo Engineering, Ishikawajima-Harima Heavy Industries, Badger, Bechtel and ICI at least have all developed pipe routing programs. In addition there have been several academic developments in various universities one of which is PIPER.

Programs which use conventional non-linear mathematical optimisation are limited to a few pipes and obstructions

[Cleveland and Close 1977 and 1979]. The other methods use heuristic approaches which will not guarantee an optimum but permit consideration of a more realistic number of obstructions and pipes. PIPER, GPS [Newall 1972, Anon 1980] and possibly others are restricted to routing planes. ISOROUTE is more general in considering the entire available space.

Bechtel's, Badger's, ICI's ISOROUTE and possibly two of the other programs can be used at the later stages of design e.g. connecting turbines or heat exchangers to steam and condensate mains which have already been defined in the data (pre-routed).

CASE STUDIES USING ISOROUTE

Design Project	Start Date	Estimated Cost £M.	Pipework Diam mm/Number/Length m		
A	1979	12	65	300	1800
B	1980	0.75	50	92	1400
C	1980	2.5	50	127	1400
D	1981	10	100	85	3000
E	1981	2.15	40	150	1300
F	1981	9.5	60	330	3750
G	1981	1.6	120	86	1500
H	1981	0.25	50	10	400
I	1981	5	185	200	4800

Table 1: Design projects using computer aids for routing pipework

Eight significant design projects (A-G and I) have used the ISOROUTE prototype as an aid to project and pipework design. Basic data about the projects is given in Table 1 above. Not all these projects have been constructed but it is possible to make some comments even at this stage.

1) Where ISOROUTE was used for estimating (projects A,B,D,E,F and I) it was regarded as extremely useful except for project I (see 5 below). Data has been transferred into the estimating and costing modules of ISOPEDAC.

2) ISOROUTE was not used until much later in project C (nearly at the detailing stage) and was unsuccessful.

3) ISOROUTE was successfully used for pipe scheming and assessing layout of equipment and pipe racks in projects A,B,D,E, and F.

4) ISOROUTE was applied without success to project G as the pipes were all routed under the tanks. This would have been acceptable in a multi-storey building, but in fact it was a tank farm. The designer omitted the ground level as an obstruction!

5) PIPER and ISOROUTE were tested on project I when assessing the impact of different layouts for the major processing units. PIPER was unable to handle any tees on the service mains and the ISOROUTE prototype could not handle enough of them to be of real value. The production version of ISOROUTE has been designed to handle data as complex as project I.

6) Project F would not have achieved the planned elapsed time without ISOROUTE. It has enabled layout decisions to be confirmed at an early stage in the project.

In summary five out of the eight projects were successful. Two of those have been very successful. Three out of the eight were not successful. One failure could have been avoided by training and the other by earlier involvement in the project. The third failure was a limitation of the prototype. The production version of ISOROUTE will handle complex tees.

CASE STUDIES USING PIPER

PIPER has been tested by several designers and engineers in ICI but used for actual design only on projects H for estimating and I for assessing site layout. The equipment for project H, and a number of routing planes were input to PDMS early in the first week. The rest of the week and part of the next were used with attempts to obtain plausible routes from PIPER.

With two days left the pipes were routed using PDMS and the estimates obtained from PDMS. Although PIPER was unsuccessful the ability to obtain estimates using PDMS was confirmed. The results for project I are described in sub-section 5 above.

OPERATION WITH THE PROGRAMS TESTED

Setting up equipment, steelwork etc. is done using cylinders or cuboids to model obstructions in the data-base. Pipe names, nominal bore, specification and connectivity must be defined but a single common specification is permissible. Obstructions are defined in PIPER by an OB/STRUCTIONS command and similarly in ISOROUTE by a TRANSFER command. PIPER does not treat

existing pipes as obstructions whereas ISOROUTE can. Both programs can be run with the ROUTE command. It is at this point that the major differences become apparent. ISOROUTE will find any orthogonal route that avoids the obstructions, whereas PIPER is restricted to the internally generated "routing planes". Pipe to pipe connections can be handled by ISOROUTE without any constraint upon the user to order the branches nor does the user need to define the branches as all belonging to one pipe. In particular ISOROUTE allows any amount of pre-routed pipework. The designer has complete freedom to use his own design and his own engineering judgement. Results are usually viewed with a graphics routine which shows the centre-lines of the pipework e.g. Figure 2.

A particular advantage of ISOROUTE is the ability to connect equipment by pipes without defining the positions of nozzles. At early stages in a project the exact positions of nozzles are rarely known. For some vessels the positions of the nozzles may be adjusted to suit the pipework. ISOROUTE will either choose suitable positions for nozzles, or use defined nozzle positions if data is available.

Another advantage is the form of pipework specification used. Normally pipework fittings are defined in a "catalogue" file. A specification then defines which fittings are to be used with a particular fluid. A typical catalogue may contain up to 100,000 items connected by over 100 specifications for a typical large project. ISOROUTE uses a simplified catalogue where fittings are defined approximately (usually as a function of pipe bore). This enables less than 10 specifications or even a single simple specification to be used.

FUTURE DEVELOPMENTS

The ability to rapidly calculate many possible pipe routes has proved very useful to project engineers. The data obtained has been used to assess layout, to estimate quantities of materials required and to estimate costs. Piping designers, however, would like to make more use of the data by editing the routes generated. This is very tedious with the alpha-numeric commands in PDMS. The key requirement is interactive graphics. When this is available it may be possible to produce the final design of the pipework by editing the routes generated by ISOROUTE. These facilities may become available in PDMS, or ISOROUTE could be linked to any of the existing interactive graphics systems available commercially.

It is also attractive to consider using CAD for extensions to existing equipment now that data can be obtained quickly from photographs [Klement and Bracewell 1982, Klement and Foord 1982].

CONCLUSIONS

"Automatic routing" and attempts to use PIPER have not been successful in ICI. There has, however, been considerable success with ISOROUTE assisting project designers. ISOROUTE does not remove the need for design skills or engineering judgement. ISOROUTE is a tool which enables a project designer to work more effectively, and has reduced elapsed time on projects. It is valued more by project designers than by piping designers.

The Critical Factors

The success or failure seems to depend at least as much on how the system can be used as upon the quality of the routes generated. Some have suggested that "preliminary piping will be done by the computer only" [Schwind 1975]. Such an approach seems to invite failure and has never worked in ICI, nor anywhere else to my knowledge. An interactive approach which assists the designer has, however, been successful in ICI, and also appears to be successful elsewhere. In particular project designers are more interested in "acceptable" and "understandable" routes, than in "optimal" routes. Piping designers, however, are more concerned with the quality of the design than the speed.

ACKNOWLEDGEMENTS

The prototype and production versions of ISOROUTE were produced by Mr W G Rees, Dr S H Boreham and Mr M H Proudler. Their developments would not have been possible without the assistance of several project engineers and designers in ICI. I would like to thank them all. I would also like to thank ICI for permission to present this paper.

REFERENCES

Allen I, Sept./Oct. 1981, ICI Magazine,59 (493) pp 104-106. "The giant jigsaw". (ICI-PDMS).

Anon, 1976, Sales literature, "Badger introduces the design machine".

Anon, 26th February 1979, Niigata sales literature, "CAD/PIPEPLAN".

Anon, 1st March 1979, Niigata sales literature, "Engineering and management via computer".

Anon, March 1980, Sales literature, "Mitsui GPS - Graphic piping design system for chemical plant engineering".

Cleveland, A B, and Close S A, September 1977, ASME publication 77-PVP-46, "Automated pipe routing for minimum cost". (Bechtel).

Cleveland, A B, and Close SA, September 1979, ASME publication 79-PVP-44, "Application of automated design to piping systems: routing and support location". (Bechtel).

Foord A G, and Rowe T S, August 1981, Pipes and pipelines International,26 (4) pp 9-16. "12 years of computer-aided piping design with ISOPEDAC".

Klement U R, and Bracewell P A, March 1982, Process Engineering,63 (3), pp 45-49, ISBN 0370-1859. "Easier routes to 'as built' records" (ICI).

Klement U R, and Foord A G, July 1982, IEE Conf. Pub. 212, Man/Machine Systems, pp 98-102, "Voice output speeds up site measurement using photogrammetry". (ICI - Photogrammetry).

Leesley, M E, and Newell RG, 1972, I. Chem E. Symposium Series No. 35, pp 2:20-2:26, "The determination of plant layout by interactive computer methods". (CADC - PIPER).

Matsui Y, et al, June 1979, IEEE publication 79CH1427-4 C, pp 121-127 "Automatic pipe routing and material take-off system for chemical plant". (Toyo).

Matsui Y, et al, October 1980, IEEE publication 80CH1511-5, pp 299-302, "Automatic pipe routing system for a chemical plant". (Toyo).

Nagai, T et al, January 1980, IHI Engineering Review 13 (1) pp 26-31, "Computer-aided general piping design system: S-BOPPS (boiler plant applications)".

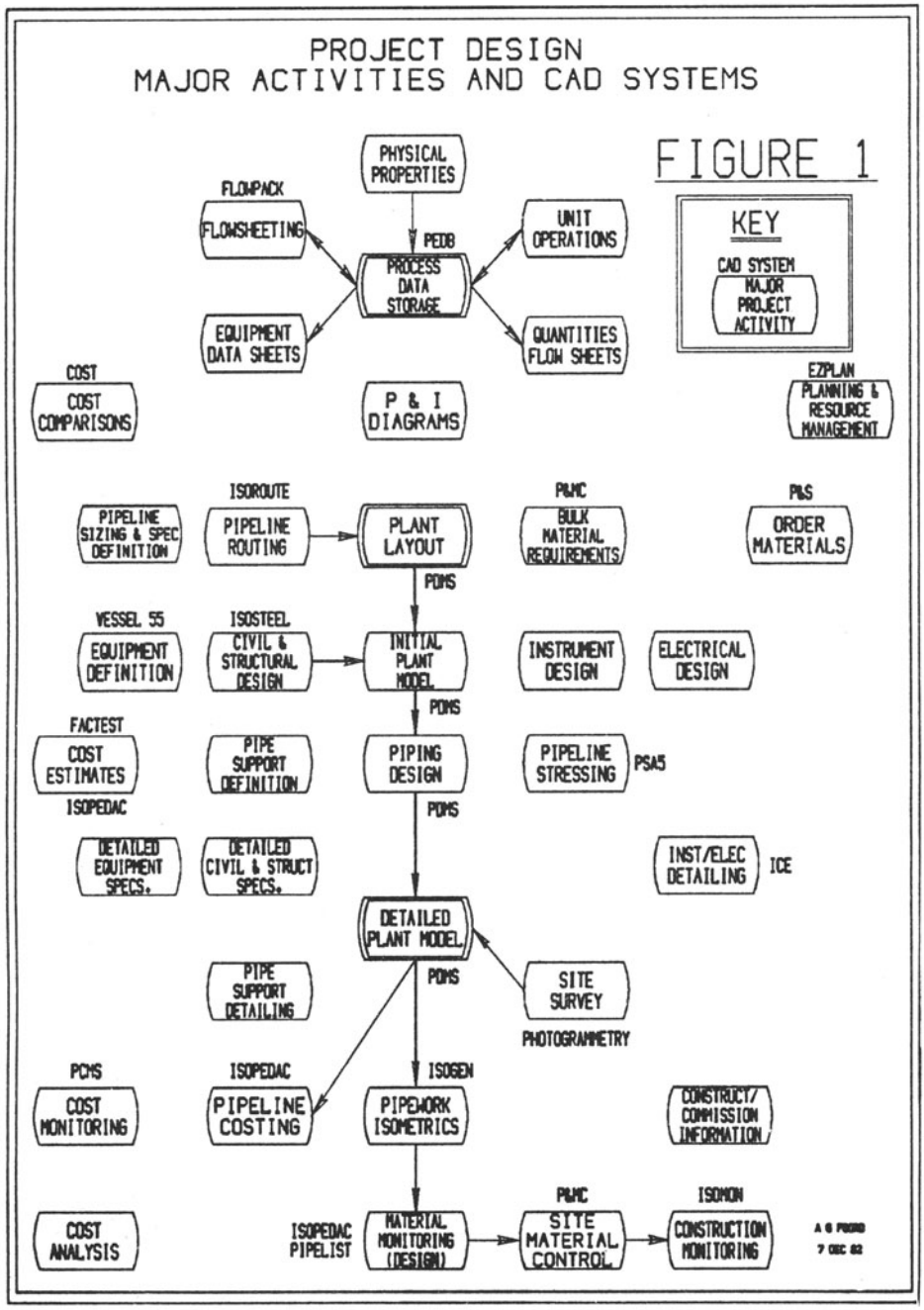
Newell R G, 1972, Information Processing,71, pp 1080-1085, "An interactive approach to pipe routing in process plants" (CADC-PIPER).

O'Shima E, and Tomita M, 31 August - 4 September 1975, Symposium COMPUTERS IN THE DESIGN AND ERECTION OF CHEMICAL PLANTS, Czechoslovakia, pp 811-821, "Algorithms for routing pipes of chemical plant". (IHI).

Schwind H, 31 August - 4 September 1975, Symposium COMPUTERS IN THE DESIGN AND ERECTION OF CHEMICAL PLANTS, Czechoslovakis, pp 787-795, "Survey on the computer-aided layout of the piping of chemical plants". (Dortmund Univ).

A G Foord/FL
16th December 1982

WP:PDMSUSERCLUB (44)


PROJECT DESIGN
MAJOR ACTIVITIES AND CAD SYSTEMS
FIGURE 1
KEY
CAD SYSTEM
MAJOR PROJECT ACTIVITY
PHYSICAL PROPERTIES
FLOWPACK
FLOWSHEETING
PEDB
PROCESS DATA STORAGE
UNIT OPERATIONS
EQUIPMENT DATA SHEETS
QUANTITIES FLOW SHEETS
COST
COST COMPARISONS
P & I DIAGRAMS
EZPLAN
PLANNING & RESOURCE MANAGEMENT
ISOROUTE
PIPELINE SIZING & SPEC DEFINITION
PIPELINE ROUTING
PLANT LAYOUT
PDMS
PMC
BULK MATERIAL REQUIREMENTS
PMS
ORDER MATERIALS
VESSEL 55
EQUIPMENT DEFINITION
ISOSTEEL
CIVIL & STRUCTURAL DESIGN
INITIAL PLANT MODEL
INSTRUMENT DESIGN
ELECTRICAL DESIGN
FACTEST
COST ESTIMATES
ISOPEDAC
PIPE SUPPORT DEFINITION
PIPING DESIGN
PIPELINE STRESSING
PSA5
DETAILED EQUIPMENT SPECS.
DETAILED CIVIL & STRUCT SPECS.
INST/ELEC DETAILING
ICE
DETAILED PLANT MODEL
PIPE SUPPORT DETAILING
SITE SURVEY
PHOTOGRAMMETRY
PCMS
COST MONITORING
ISOPEDAC
PIPELINE COSTING
ISOGEN
PIPEWORK ISOMETRICS
CONSTRUCT/ COMMISSION INFORMATION
COST ANALYSIS
ISOPEDAC PIPELIST
MATERIAL MONITORING (DESIGN)
PMC
SITE MATERIAL CONTROL
ISOMON
CONSTRUCTION MONITORING

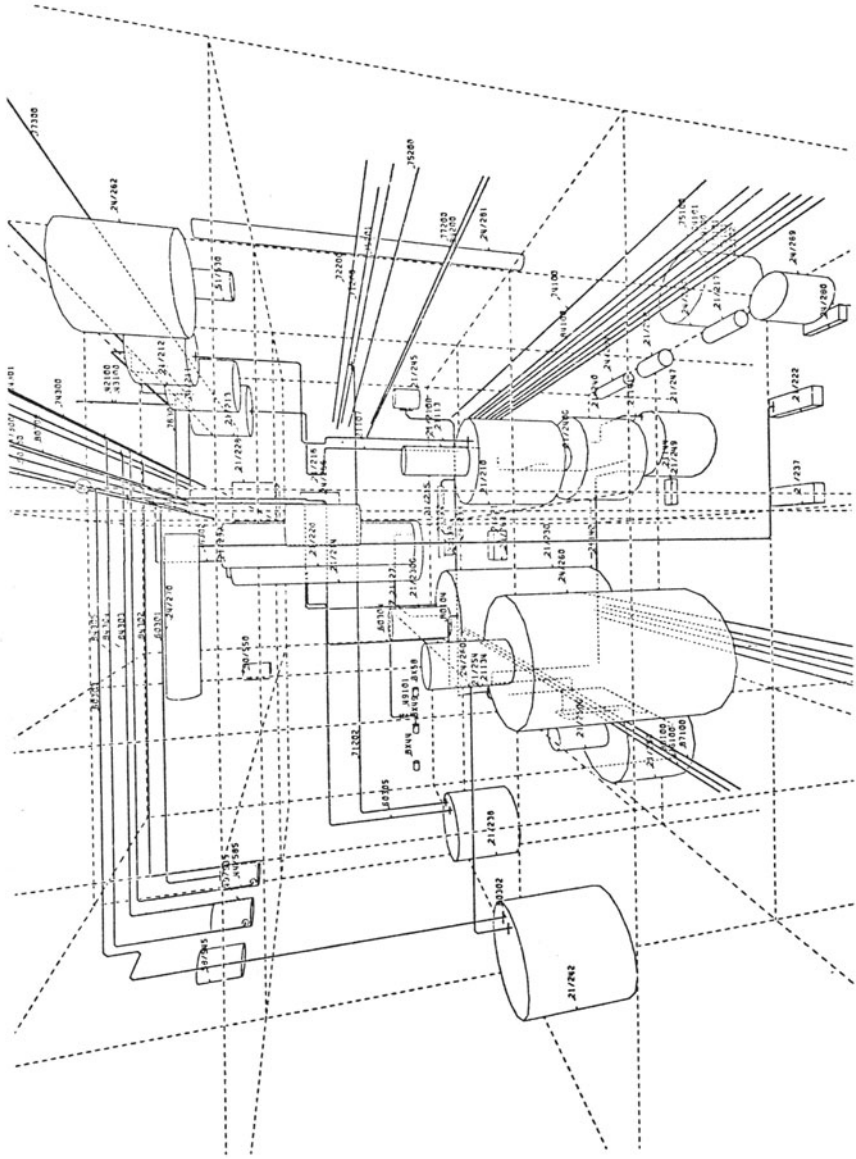


Figure 2 Perspective view of part of Project A

DATABASES

THE FURTHER DEVELOPMENT OF A GEOTECHNICAL/GEOLOGICAL DATABASE

L.A. Wood*, E.V. Tucker** and R. Day**

* Dept. of Civil Engineering, Department of Geological Sciences, Queen Mary College, London.

SUMMARY

The continued development of the GEOSHARE database system for the storage of geotechnical and geological records is discussed. At the moment progress has been concentrated upon the use of an ICL 2980 mainframe installation, although in the longer term complementary systems for both mini- and micro-computers are envisaged. The object is to combine the advances in word processor and computer technology in order to provide a system that is simple to use, cost-effective and is of immediate relevance to both the Professional Civil Engineer and the Geologist. Examples of the use of the pilot system are given in relation to a geological study of part of the Thames Estuary.

INTRODUCTION

The projection of a surface onto a two-dimensional sheet of paper involves few major difficulties as illustrated by the excellent topographical maps compiled by the Ordnance Survey. Geological maps on the other hand suffer from a major handicap in that a three-dimensional solid is transcribed onto a two-dimensional plane with the loss of the third dimension. Further complications arise due to the self-evident fact that many features of interest are buried beneath the surface details of which may only be obtained from drilling operations. A situation exists therefore where it is very expensive (in comparison with topographical mapping) to obtain the information and yet upon receipt it is not possible to present it to best advantage due to the difficulty of projecting a solid onto a sheet of paper.

The expense incurred in drilling boreholes may be somewhat alleviated, or perhaps more realistically the instance of sampling increased, if the wealth of data already contained within site investigation reports (also termed ground investigation

reports) could be fully utilised in the preparation of geological maps. An awareness of the value of general geological information contained in these reports has been highlighted in recent years by the discussion generated within the industry following the publication of the CIRIA report "A case for a National Registry of Ground Investigation Reports" (1977). This has lead to the adoption by CIRIA, in conjunction with the Institute of Geological Sciences of the proposal to establish a central library of reports stored on micro-fiche. Although some misgivings are felt by practitioners as to the establishment of such a library there is no doubt that the factual information contained within the reports may be viewed as part of the national heritage. Any suggestion that it would lead to a general decline in the number and quality of site investigations is considered to be without foundation, indeed the reverse might be true especially in terms of quality. The establishment of the Ordnance Survey did not see the end of the private land surveyor and is now accepted as of great social and economic benefit to the whole nation.

The creation of general, truly three-dimensional geological maps has become possible with the advent of computer technology. The combination of a computerised databank of geological information, primarily drawn from borehole records, coupled with easy to use routines allowing immediate access to geological sections, isopachyte maps, shear strength profiles, contour maps of subsurface horizonts and so on; would provide both the geologist and the civil engineer with a most formidable aid with which to facilitate his interpretation of local ground conditions within his sphere of interest (Wood, 1981).

The concept of such a databank is not new and a number of systems have been devised for commercial and institutional use (e.g. Loudon, et al. 1977). However, these systems have not been widely adopted because in part they rely on coding and the use of mnemonics that must be understood by the operator, or they require extensive checking on input to ensure the accuracy of the information stored. The excessive cost involved in these operations prohibits the development of an economically viable system because of the large volume of data that should be stored.

The GEOSHARE system (Wood et al. 1981) avoids the use of external codes or mnemonics and instead accepts data in English; utilising the limited vocabulary associated with geotechnical/geological information and performing invisible, internal coding for efficient storage. Indeed ease of operation has been recognised as paramount and the following objectives have been defined:-

(a) the system must be secure and hold factual information only;

(b) the factual information must be as complete and accurate as possible, the position of a borehole must be defined both in plan and level, for example, to be acceptable;

(c) as full a description as possible of the different strata, must be stored;

(d) provision must be made for the storage of information relating to mechanical properties, water level readings etc.;

(e) the language of operation of the system is English, machine defined mnemonics are not used, although the facility for temporary user defined mnemonics would be provided;

(f) extensive logical error checking of all the input must be undertaken; for example, the succession entered would be automatically checked against available information in the store and queried (not rejected since final control rests with the user) if substantially different;

(g) data input must be both fast and simple, with a variety of modes ranging from "question and answer", through pro-forma to digitiser and tablet operation;

(g) retrieval must be both fast and simple, with a large variety of styles of presentation ranging from the simple printed text (borehole log) to sophisticated graphical forms when necessary;

(i) throughout control must rest with the user and interpretation of the data must also rest solely with the user;

At present GEOSHARE has met most of these objectives but further work is still required in order to refine the system and improve its robustness, reliability and efficiency in certain areas to meet commercial standards.

In order to test the use of such a system and to enhance its development within a user environment, GEOSHARE has been employed in a geological assessment, based upon some 400 boreholes logs, of an area from the north bank of the River Thames to Maplin Sands. Results obtained from this study are given below in order to demonstrate certain of the advantages that would accrue to the practitioner from such a system.

DATA STORAGE SYSTEMS

Previous

Various systems for storage and retrieval of data have been tried in the past, based largely on a fixed format or 'pseudo-free' format approach and using punched-card technology, implemented in a 'batch' processing situation. These factors limit the extent to which a user orientated environment can be developed.

Buller (1964,1972), Harvey (1973) and Loudon et al. (1977) utilised fixed format systems coding data by reference to a coding manual prior to input. Pre-input coding not only requires much effort, an important consideration when storing voluminous archive material (Farmer and Read, 1976) but gives little room for flexibility when inputting data and can introduce errors.

Both fixed and 'pseudo-free' format techniques were used by Rhind and Sissons (1971), the former consisting entirely of numerical characters, information not in numeric form being trnasformed by means of codes. The 'pseudo-free' format section used normal English modified, for example, by placing characters other than primary constituents in parentheses. A dual format approach was adopted by Gover et. al. (1971) making use of a 202 word vocabulary against which input words were checked automatically and coded, if present. Field trials by geologists (Farmer and Read 1976) indicated the advantages of the 'pseudo-free' format approach because no fundamental changes in methods of recording data were required. The limitations of the vocabulary proved frustrating at times when new phenomena needed to be described. The system was regarded as viable only if archive material of more 30 000 records could be stored but the cost of effective supervision of the staff inputting data, outweighed the benefits of the system.

Cripps (1978a, 1978b, 1979) found that due to the restructed range of sediments encountered in a project based on Newcastle-Upon-Tyne, a three digit number and an optional twelve character expression was adequate to describe a single stratum prior to input. The character expression gave flexibility to the data stored but the method suffered from the same disadvantages as other non-English input systems.

Simplicity of operation is the key factor if a system is to be adopted by industry. The Saskatchewan Government Well Data System (Buller, 1972) incorporated a retrieval method designed for use by geologists without previous programming training but the objective was not borne out in practice and training in both programming and computing experience was found to be necessary. An interactive system designed around

standard English terms and operable by semi-skilled personnel is essential for the widespread acceptance and use of a data bank.

Gover et al. (1971), Harvey (1973) and Loudon et al. (1977) all used retrieval commands in near English but the commands must be learned before a retrieval can be made. Cripps (1979) devised a retrieval system which works in response to a series of programmed interactive questions enabling the user, with only limited computer experience, to communicate with the computer in standard English, precluding the need to learn data control instructions.

GEOSHARE extends the method of Cripps so that data, on input, can be typed directly into the computer in a single operation as well as being retrieved in response to a series of programmed questions. This dispenses with the costly procedure of recording data on punch-cards prior to input. Coding of data is performed internally by the computer and does not involve the operator. Flexibility of terms on input is achieved by a self updating vocabulary producing a simple to operate system.

GEOSHARE

To maximise the utility of the system external coding and mnemonics are avoided and operational simplicity is obtained by inputting data in standard English in response to a series of programmed questions. It should be emphasised that alternative input routines, such as the presentation of a form to the user are also under development. The data is coded internally by the computer as well as deciphered when accessed.

A twofold division of information present in the Site Investigation borehole record can be made:-

(a) Numerical data.
(b) Descriptive data.

The former includes numerical borehole reference data, sample tests and water level data. The latter includes soil description and non-numeric borehole reference data as shown in Table 1. Space for general comments exists, up to a maximum of fifty characters per borehole. Category (a) contains items esstially numeric in character that require no coding in the computer. Some of this data will be of a type that can be checked automatically by the computer for logical errors, on input. For example, if the day of the month is input as 32 a correction will be called for by the computer. Non-logical errors can only be found by manual checks and the numeric borehole data is displayed after input at which point. Mis-

Numeric Data

Borehole Reference data	:	grid reference
		ground level w.r.t. Newlyn
		date of drilling
		borehole diameter
		casing limits
Sample and test data	:	depths to strata
		in-situ tests (such as SPT)
		sample recovery
		laboratory tests
		water level readings from standpipes etc.

Descriptive Data

Soil descriptions	:	consistency/compaction
	:	colour
	:	structure and organic content
	:	primary soil type
	:	secondary soil type
	:	formation name/geological origin
General Comments	:	50 characters per borehole

Table 1 Stored information

takes can be identified and corrected. A correction facility at this stage is important because it gives the operator an opportunity to recognise an error on input that might otherwise be overlooked.

The descriptive data of category (b) involves terms and phrases having a particular bulk, dependent on the completeness of the soil description and coding of this information is necessary for efficient storage. A scheme has been devised whereby data is presented to the computer in normal English, to create a user friendly environment, and the coding operation is performed internally by the computer. To achieve this a series of vocabularies are created, each one serving a specific part of the borehole record description. This principle applies to all categories of the non-numeric data that need to be identified but the concept will be illustrated by reference to the vocabularies designed for the soil descriptions.

The Pleistocene and Holocene soils of the Thames estuary can be adequately described by reference to six qualitative headings:- colour, primary soil type, secondary soil type (or formation name if known), soil structure (including organic content), consistency, compaction. These headings form convenient divisions in which vocabularies of relevant terms can be stored.

At the moment on input the operator is asked to provide information relating to each division (category) and responds to each programmed question using normal English. Either the complete term or a truncated form comprising the first four letters are entered to a maximum of three terms. This limitation on the number of terms is employed because longer descriptions are uncommon, and if used, normally result in a loss of precision. The terms are checked by the computer against the appropriate vocabulary and, if present, coded and stored. If a term is input that is not present in the vocabulary the two closest alphabetical terms are displayed to the user as possible alternatives. This device draws attention to possible spelling or typing errors and provides a ready means of correcting mistakes. The operator can either accept one of the alternative terms, re-input the data or update the vocabulary file.

Retrieval of information is also in response to a question and answer routine; data being retrieved by area, by borehole or by stratum horizon. A suite of programes has been developed including: complete listings, depth to formation boundaries, depth to occurrence of a specified soil type, soil type at a specified depth and the identification of different types of cyclic sequences. Output data may also be processed and linked to programs developed by others to perform tasks such as contouring (Wood et. al. 1981). The examples detailed below

have been chosen in order to demonstrate the versatility of such a system as an effective interpretative tool.

GEOLOGICAL STUDY

The present study is based on sites within the River Thames estuary where mixed sequences of Pleistocene and Holocene superficial deposits overlie a highly irregular London Clay surface. Other Teriary rocks are present locally. In channel areas the superficial deposits reach 40m in thickness and show a relatively complex sequence of alternating gravels, sands and clays. The thickness away from channels averages 16 to 20m and successions are comparatively simple, with a tripartite division into basal gravels (partly of Pleistocene age) overlain by silts and clays followed by cover sands. A refined sediment analysis has been possible for sites at Maplin Sands and Foulness Island (Greensmith and Tucker, 1971) using criteria that include lithification, cohesion, sediment cyclicity and fauna, showing that a number of phases of emergence of the depositional surface followed by transgressive submergence have occurred during the Holocene. Identification of levels of emergence by direct reading of bore logs is rendered less exact by the general nature of soil descriptions but progress towards the recognition of these levels, where soils are more consolidated, has been achieved by computer analysis.

Stratum analysis and identification of formation boundaries

The formation is the fundamental unit in lithostratigraphy and a prime requirement in geology and soil engineering is to establish the boundaries that separate formations within which lithological unity exists, to facilitate the drawing of a geological section. Many formation boundaries are gradational and are not represented by a sharp lithological change from one soil or rock type to another. The position at which a boundary is drawn can be subjective and traditionally will be decided by a geologist by reference to his experience of the stratigraphy of the area. The computer on the other hand must be programmed according to a well defined methodology. This has the disadvantage of losing 'geological intuition' but the advantage of repeatability. Accuracy can be maintained by quantifying the instructions to an extent that would be impracticable in manual procedures.

The adopted routine first differentiates between the basement beds and the overlying strata. Formation boundaries within each relevant borehole are then determined in accordance with the percentage of a particular soil type present within a series of overlapping blocks (Day et. al. 1983) with regard to that borehole alone. These boundaries are expressed as a percentage of the depth to the basement beds, which act as a marker horizon, and compared with neighbouring boreholes.

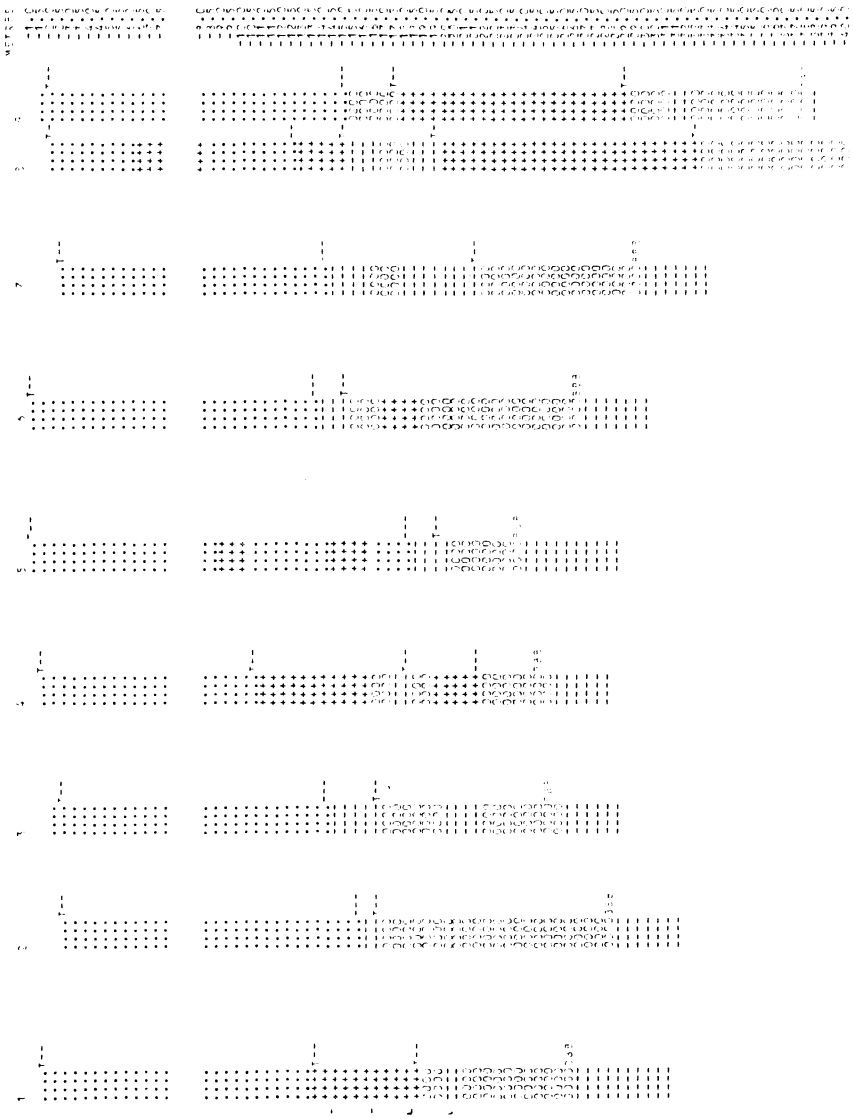


FIG. 1 GEOLOGICAL SECTION

SN	GRID REF	FORMATION BOT	FORMATION TOP	AMOUNT ACTUAL	AMOUNT EXPECT	SOIL
TR	31 859					
		-13.05	-12.95	0.03	0.01	SILT
		-14.65	-1.25	0.42	0.22	SAND
		-26.75	-19.05	0.27	0.10	GRAV
TR	37 869					
		-15.10	-2.40	0.43	0.23	SAND
		-27.30	-16.20	0.29	0.13	GRAV
TR	43 875					
		-13.50	-2.00	0.43	0.23	SAND
		-25.00	-15.70	0.28	0.11	GRAV
TR	49 883					
		-21.30	-10.20	0.11	0.05	SILT
		-10.30	-1.20	0.37	0.19	SAND
		-24.30	-17.60	0.25	0.10	GRAV
TR	56 890					
		-17.46	-2.60	0.43	0.33	SAND
		-27.10	-19.80	0.16	0.05	GRAV
TR	62 899					
		-17.30	-2.30	0.40	0.27	SAND
		-36.40	-14.35	0.39	0.19	GRAV
TR	66 905					
		-21.30	-13.75	0.06	0.03	CLAY
		-13.70	-2.25	0.79	0.33	SAND
		-29.75	-21.50	0.35	0.25	GRAV
TR	74 913					
		-13.10	-14.80	0.01	0.00	CLAY
		-32.50	-13.45	0.11	0.04	SILT
		-13.20	-1.85	0.37	0.17	SAND
		-40.40	-32.30	0.25	0.17	GRAV
TR	75 917					
		-25.35	-17.15	0.10	0.04	SILT
		-14.70	-1.50	0.25	0.11	SAND

---- ---- ----	CLAY	CCCC CCCC CCCC	CHALK	TTTT TTTT TTTT	TOPSOIL
++++ ++++ ++++	SILT	PPPP PPPP PPPP	PEAT	MMMM MMMM MMMM	MADE GROUND
....	SAND	SSSS SSSS SSSS	SHELLS	LLLL LLLL LLLL	LONDON CLAY
0000 0000 0000	GRAVEL				

--- DEFINES LIMIT OF A FORMATION THAT OCCURS AT THAT X DEPTH TO BASEMENT IN GREATER QUANTITIES THAN WOULD BE EXPECTED IF THE SEDIMENT DISTRIBUTION T-- WAS RANDOM. T-- INDICATES TOP.

BBP DEFINES BASEMENT TOP

FIG. 1 GEOLOGICAL SECTION (CONT.)

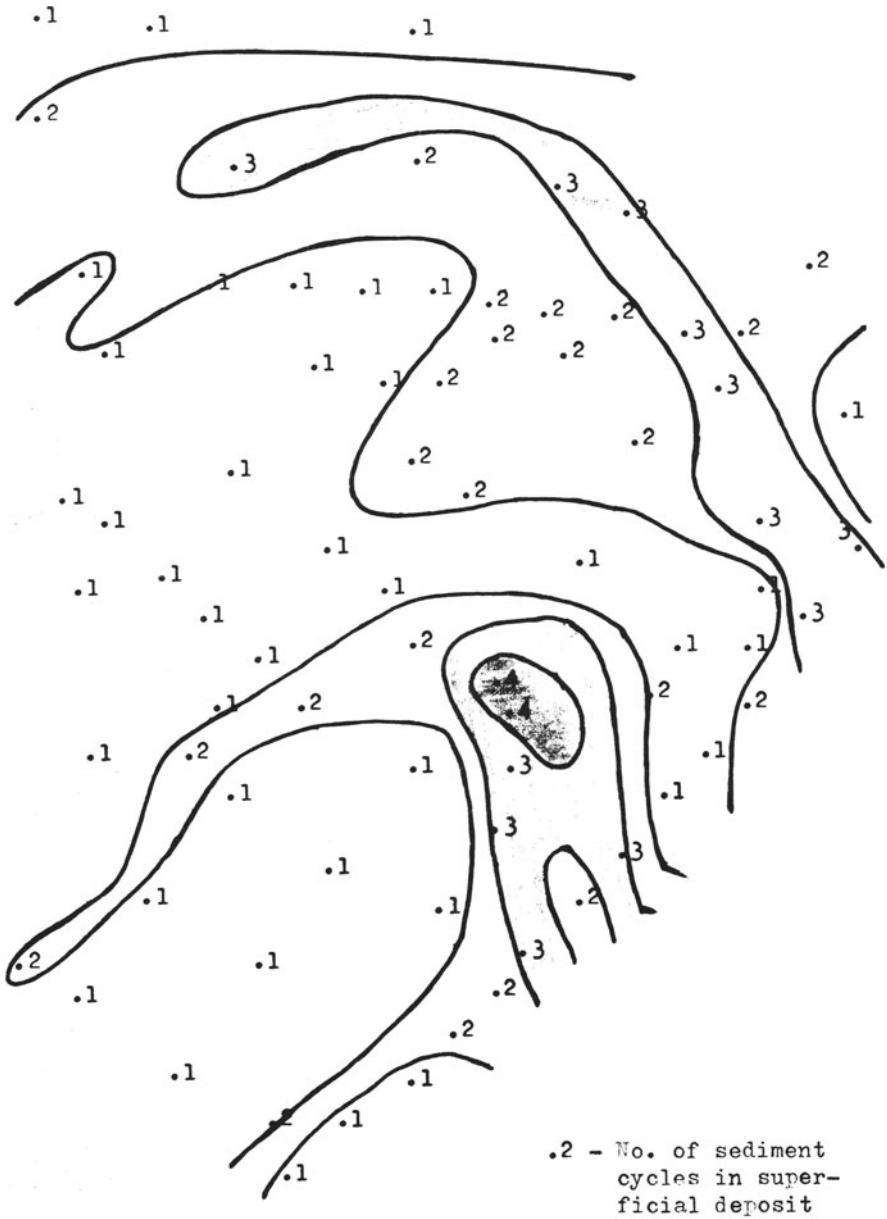


FIG. 2 CONTOUR MAP OF CHANNELS IN THE LONDON CLAY

Thus regional trends (the extent of the region is determined by the user) are taken into account in the final analysis of the formation boundaries. A typical section obtained from such an exercise and originally output as a line printer character plot is shown in Fig. 1.

The identification of channel and interfluve successions

In south-east Essex superficial deposits overlie the London Clay basin which is known to have an irregular morphology due to the incission of channel cutting streams. Identification of the extent of these buried channels is of major importance to both geologists and civil engineers. Structure contour maps of the basement surface provide one means of channel identification, but often the issue is complicated by a regional surface trend. This may be resolved by the use of a trend surface analysis routine and subsequent contouring of the residual values (Wood et. al. 1981). An alternative approach is to consider the nature of the overlying Sedimentary Succession. A higher environmental energy is found within a channel than within an interfluve region, and this may be reflected in the grain size of the deposits. Evidence for emergence should be found in both environments, but an interfluve might show fewer phases and each phase would be thicker. A channel area would be more sensitive to relative variations in sea level and this would be reflected in the number and cyclic sequences and the number of lithological changes persent. The effect of erosion, particularly by scouring within the channels, will however play an important role, and may blur the Sedimentary differences between channel and interfluve sequences.

A typical contour map based upon the number of coarsening-up cycles pertaining at borehole locations is shown in Fig. 2 clearly illustrating the presence of channels.

CONCLUSIONS

The pilot study has demonstrated the feasibility of storing and retrieving Site Investigation borehole records in both a fast and simple manner, utilising interactive computer techniques. In addition the value of such a system in aiding interpretation of geological/geotechnical data has been proved. Extension of the system to other geological settings and geographical areas could be achieved with very little modification. To date the system has been written in Fortran in order to ensure portability; but before any further development takes place this decision is to be reviewed. In addition although the question and answer style of the input routines has proved acceptable it is anticipated that future work will concentrate on other, more efficient, forms of input based upon word processor technology.

At this stage, with the pilot study complete, the feasibility and desirability of the use of a computerised databank, such as GEOSHARE, has been clearly demonstrated. The next, more difficult stage is the realisation of the commercial potential and exploitation of such a system.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. R.A.B. Bazley of the Institute of Geological Sciences and Binnie and Partners for making available the borehole logs on which this study is based. R. Day acknowledges the financial support of the Science and Engineering Research Council.

REFERENCES

Buller, J.V. (1964) A Computer-Oriented System for the Storage and Retrieval of Well Information. Bull. Con. Pet. Geol. 12, Part 4, pp. 847-891.

Buller, J.V. (1972) Development of the Saskatchewan Computerized Well Information System, 1964-1971. 24th Int. Geol. Congr. Section 16, pp. 97-102.

Cripps, J.C. (1978a) Computer storage of site investigation data for tunnelling. Computer Methods in Tunnel Design. Institution of Civil Engineers, London, pp 183-196.

Cripps, J.C. (1978b) Computer geotechnical data handling for urban development. Proc. III Int. Cong. Int. Ass. Eng. Geol., Madrid, Spec. Ses. 4, pp 147-154.

Cripps, J.C. (1979) Computer Storage of Geotechnical Data for Use During Urban Development. Bull. Int. Ass. Eng. Geol., No. 19, pp 290-299.

Day, R.B., Tucker, E.V. and Wood, L.A. (1982) The computer as an interactive geotechnical databank and analytical tool(In press).

Farmer, D.G. and Read, W.A. (1976) A Minimal Effort for Maximal Return Philosophy Applied to IGS Onshore Borehole Records. Computers and Geosciences 2, pp. 365-374.

GOVER, T.N., Read, W.A. and Rowson, A.G. (1971) A pilot project on the storage and retrieval by computer of geological information from cored boreholes in Central Scotland. Inst. Geol. Sci. Rep. No. 71/73: 30.

Greensmith, J.T. and Tucker, E.V. (1971) The effects of late Pelistocene and Holocene sea-level changes in the vicinity of the River Crouch, East Essex. Proc. Geol. Assoc. London, 82,

Part 3: 301-322.

Harvey, B.I. (1973) A Computer System for Storage and Retrieval of Hydrogeological Data from Well Records. Rep. Inst. Geol. Sci., No. 73/18, 34pp.

Loudon, T.V., Cubitt, J.M., Henson, M.R. and Duncan, S.V.(1977) South Essex geological and geotechnical survey, Part 4: Computing. Inst. Geol. Sci.

Rhind, D.W. and Sissons, J.B. (1971) Data banking of drift borehold records for the Edinburgh area. Rep. Inst. Geol. Sci., No. 71/15: 19pp.

Tuckwell, D.J. and Sadgrove, B.M. (1977) A Case for a National Registry of Ground Investigation Reports. CIRIA, Report 70.

Wood, L.A. (1981) A data base of site investigation reports. Ground Engineering. 13, No. 6: p2.

Wood, L.A., Tucker, E.V. and Day, R.B. (1981) GEOSHARE : The development of a databank of geological records. Engineering Software II, Pentech Press. (Also to appear in Advances in Eng. Software).

A MICROCOMPUTER ORIENTATED MATERIAL PROPERTY DATABASE FOR PROCESS ENGINEERING AND DESIGN

P. Norman

Department of Chemical Engineering,
University of Newcastle upon Tyne.

INTRODUCTION

The present package consists of a physical property database and generator and is the first step in the construction of a wide-ranging material property database of interest to the Process Industries. It is largely based on the work of Reid and Sherwood[1977] and is written entirely in ANSI FORTRAN IV. Therefore it is readily mounted on any machine with a FORTRAN compiler. The package was originally intended for use on microcomputers and in that context is close to the maximum size for a 64K, 8-bit micro. This has meant that some desirable functions have not been implemented. Several possible strategies for increasing the number of functions and for extending the scope of the database are presented later. It has been called a Material Property Database as it is intended to extend the package to include text information such as toxicity and methods of handling.
The package is an interactive design tool and provides the user with data about pure components or mixtures of up to 20 components, usually at a specified temperature and pressure. It provides process engineers with quick assessments of physical properties. It is also possible to interface the package directly with a user's programme written in FORTRAN.
The user is guided through a question and answer session to establish his requirements and then the necessary data is retrieved from a database or generated by the programme. The size of the database is virtually unlimited. No computing skill is required to use the package. Materials and properties are

identified by code number obtained from a handbook.

STRUCTURE OF THE PHYSICAL PROPERTIES PACKAGE.

There are 26 properties available for pure components or mixtures. These include:

a) for mixtures or pure components at a specified temperature and pressure -
    vapour and liquid enthalpy
    specific heat
    viscosity
    density
    vapour fraction, etc.
b) for a mixture - bubble and dew point.
c) for pure components at a specified temperature -
    vapour pressure
    boiling point.
d) pure component constants -
    critical data
    flash point
    normal latent heat, etc.

Mixture property data is constructed from pure component data which is experimental or may be estimated. The pure component data is contained in the DATABASE, which is virtually unlimited in size, while the data construction is undertaken by a set of PHYSICAL PROPERTY SUBROUTINES. The whole exercise is supervised by an EXECUTIVE programme. A block diagram of the structure is presented in Figure 1.

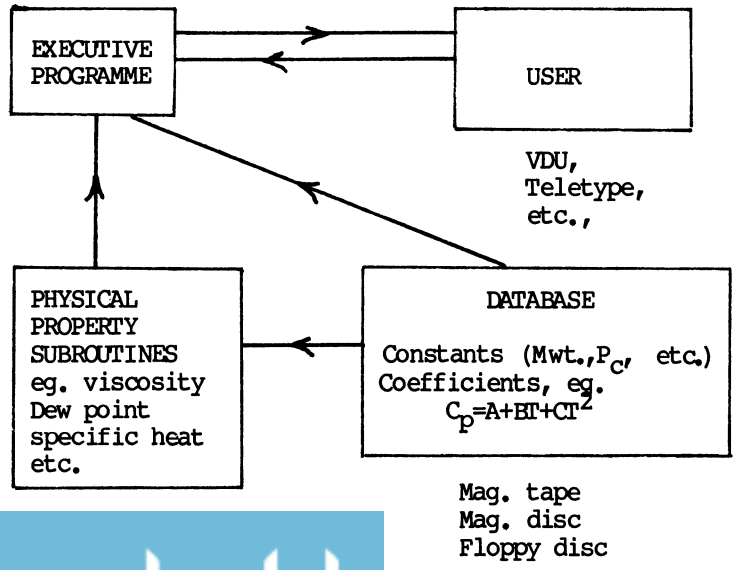


Figure 1. Structure of Physical Property Package.

The EXECUTIVE

This is the main programme and its function is:

a) establish the user's requirements
b) load the necessary data from the database.
c) invoke the appropriate property subroutines.
d) display the results to the user.

The user's requirements are determined in a short question and answer session to which the user replies with Y(yes) or N(no), code numbers for the components and property required and the temperature, pressure and composition of the material. Code numbers are obtained from a handbook. It would be possible to devise an on-line 'help' system rather than rely on printed material but this is not practical within the limited space of an 8-bit micro.

Having decided what is required, the executive loads data from the database using a data loading subroutine (BLOAD). The property subroutines are then called as required by the calculation in hand and finally the data is displayed. The results can be stored in a table for later display so that, for example, the change in vapour pressure as a function of temperature can be plotted over a temperature range.

There is a facility built into the executive to allow the user to enter physical property data from the keyboard that will overwrite some, or all, of the data loaded from the database. This is useful where the user might have some accurate experimental data for one of the components in a mixture and wishes to use that in preference to the stored data.

In recognition of the fact that correlations are only approximations and apply over limited ranges, or that estimations may only apply under some conditions the executive allows the user in certain circumstances to call an alternative method for calculating the required result. Again, because of space limitations, this has not yet been fully exploited.

The block diagram for the executive is shown in Figure 2.

Data loading Data is arranged in the database as blocks of 10 lines, each containing 8 data items. The data can be dumped to a printer and inspected if required. The first line of each block contains the name of the component; the next 9 lines contain the data. The main programme allocates an array of 180x8 to hold a maximum of 20 blocks of component data while the names are stored in a separate character array. This represents a 6K overhead in the programme and if space is limited some could be reclaimed by reducing the number of components that can be handled.

The data loader (BLOAD) asks for the component

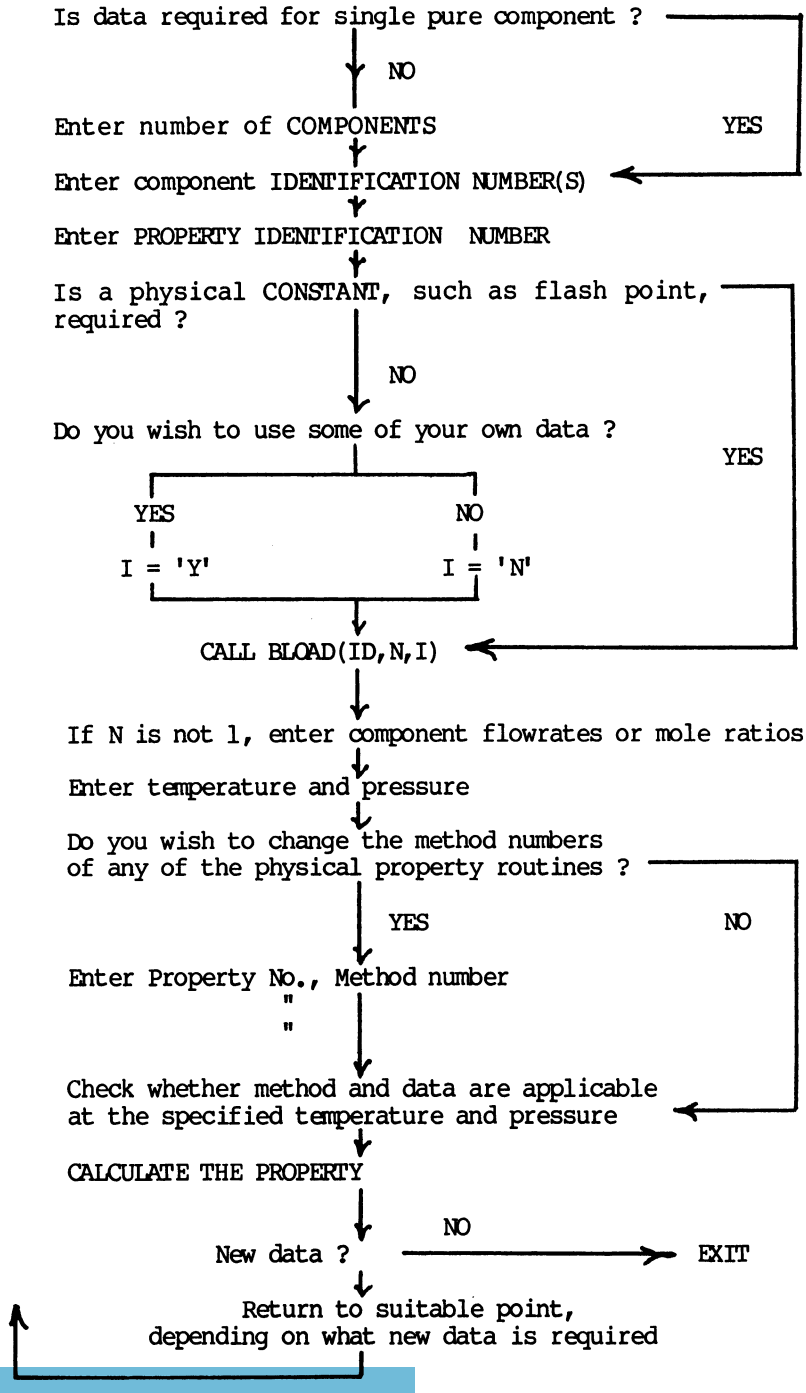


Figure 2. Structure of the executive.

identification numbers (ID). These can be given in any order. Before reading the database, BLOAD reorders the IDs in ascending numerical order. This is so that reading of the database is made faster; to reach the data for component number N, the database was originally rewound after reading data for the previous component and the programme then had to read and discard the first character of 9N data lines before block N was reached. With the IDs reordered, the programme can now scan sequentially through the data blocks. This means the database is read once only from the beginning to the highest ID requested. This has lead to quicker access and also means that it is practical to use magnetic tape to hold the data. The user, of course, also has the convenience of being able to enter IDs in arbitrary order.

Once the data blocks have been read into memory, the components (now in their ordered sequence) are then referred to by their internal storage number; ie., if the user has specified components 9,33,2,17, they are reordered by BLOAD as 2,9,17,33 and then, when in memory, are referred to as 1,2,3,4.

Although the user can give the IDs in any order, the component flows or ratios must be given in the same order. Since BLOAD has reordered the IDs, the flows must also be reordered. This is achieved with subroutine SORTFL.

The structure of the database is shown in Figure 3.

```
line no.

   1    NAME OF FIRST COMPONENT (ID 0)
   2    <  8 ITEMS OF DATA  >
   3             "
   4             "
   5             "
   6             "
   7             "
   8             "
   9             "
  10             "
  11    NAME OF SECOND COMPONENT (ID 1)
  12     <  8 ITEMS OF DATA  >
  13             "
  14             "

      ETC.
```

Figure 3. How the data is organised on the database storage device.

This method of storing data was chosen to avoid many of the problems of machine dependence with more efficient storage schemes.
The structure of BLOAD is shown in Figure 4.

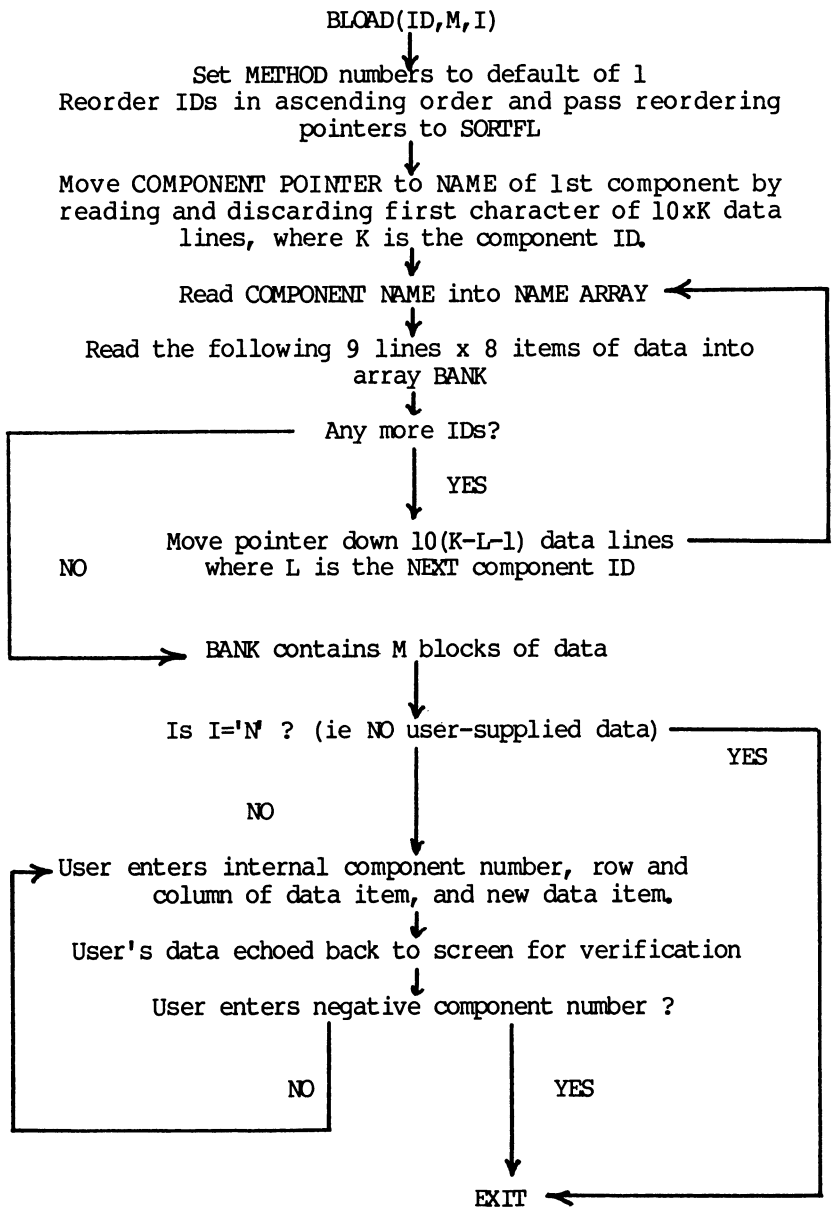


Figure 4. Structure of data loading subroutine, BLOAD

THE PHYSICAL PROPERTY SUBROUTINES

The executive calls on the subroutines when necessary to construct the required information. The procedures used are briefly outlined below.
The following pure component constants are read directly from the array BANK:
Molecular weight, the critical constants, normal boiling and melting points, the normal latent heat, and the acentric factor. Flash point and flammability data is also avialable.

Vapour pressure: If $T>T_c$ Henry's Law solubility coefficients are used. Otherwise, a check is made to see if the specified temperature is outside the range for Antoine correlation. If it is not, the coefficients for a 3-term Antoine correlation are obtained from the BANK array. If the temperature is out of range, vapour pressure is estimated using Riedel's method and other appropriate data in the array.

Vapour specific heat, Cp^o: The mixture or pure component is treated as an ideal gas. Specific heat is calculated for each component from a 4-term polynomial, the coefficients being obtained from the BANK array. A mixture specific heat is obtained from mole fraction averaging of the pure component values.

Vapour enthalpy, HV: The mixture or pure component is treated as an ideal gas, with the datum HV=0 at 298K. The Cp^o coefficients are obtained from BANK and used in the integrated specific heat equation. A mixture enthalpy is obtained by mole fraction averaging.

Liquid enthalpy, HL: This is related to the vapour enthalpy datum (HV=0 at 298K). The mean liquid specific heat is calculated between the temperature of interest and the normal boiling point. Liquid enthalpy is then the vapour enthalpy at the normal boiling point, less the normal latent heat and the heat change of the liquid in reducing its temperature from the boiling point to the temperature of interest.
If any component is above its critical temperature, it is assumed to be a gas dissolved in the liquid and the enthalpy of that component is calculated accordingly.
A mixture liquid enthalpy is obtained by mole fraction averaging of the pure component liquid enthalpies.

Liquid specific heat, S: This is formed from a 3-term polynomial correlation of experimental or predicted data, in a similar fashion to vapour specific heat.

Pure component boiling point as a function of pressure: The routine determines the temperature at which the vapour pressure equals the specified pressure.

Density, RHO: Density can calculated for a liquid or a vapour. In this case, the problem is 'set up' in the executive. A phase split routine is called first to determine whether the specified mixture is a liquid or vapour or both. If the mixture contains both phases, a warning message is printed and the user is advised to determine the composition of each phase and to do separate density calculations on them.
For a liquid mixture, a routine is called which applies the Francis equation, corrected for pressure. This relationship is based on a correlation of experimental data. If Francis coefficients are not available, density is estimated by a routine using the Gunn-Yamada-Rackett method. The mixture density is obtained from the pure component densities by mole fraction averaging.
If the specified material is a vapour, the density is determined using the Lee-Kesler generalisation of the Benedict-Webb-Rubin equation of state. With pure components, this can be applied direct but if the material is a mixture, a routine is first called that sets up the pseudocritical constants for the mixture.

Vapour viscosity, VISCV: The technique used for calculating viscosity depends on the specified conditions and, as for density, the problem is 'set up' in the executive. The executive first determines whether the material is a mixture or a pure component.
A] MIXTURE: The Childs-Hanley test is applied to determine whether the mixture is high or low pressure. If it is high pressure, a correction is determined using Dean and Stiel's method. Next, the low pressure viscosity is determined for each component in the mixture and the low pressure mixture viscosity is then derived using Brokaw interaction parameters. If necessary, the high pressure correction is then added.
B] PURE COMPONENT: The same routine is used as for the low pressure pure component viscosity in the mixture case above, except that the Childs-Hanley test is applied to the pure component. If it is high pressure, a correction is determined using the Jossi-Stiel-Thodos method. In this routine, for both high and low pressure cases, allowance is made for a component being polar.

Pure component latent heat as a function of temperature, LAMT: This is determined using the Watson equation.

Bubble point and Dew point, BBUBPT,BDEWPT: These are determined in the conventional manner from the specified mole fractions in the mixture, the pressure and the vapour/liquid partition coefficients (K values) for each component. The accuracy depends on the method of evaluating the K values. At present, because of limitations on computer memory, the only K value option is for ideal mixtures.
The vapour pressure of each component is determined (as described above) and divided by the specified pressure to yield the K values.

Phase determination, BPHASE: For a specified mixture composition, temperature and pressure, the vapour fraction is returned. A check is first made to determine if the material is a pure component. If it is, vapour fraction is returned as 1 or 0 depending on the temperature and pressure. If the material is a mixture, a check is made for any supercritical components. If any are present, these are treated as dissolved in a reference fluid (water) for the purposes of calculating the phase split. As for the bubble and dew points, the accuracy depends on the method of estimating K values (whether for a dissolved gas or condensed vapour). Again, the mixture is treated as ideal.

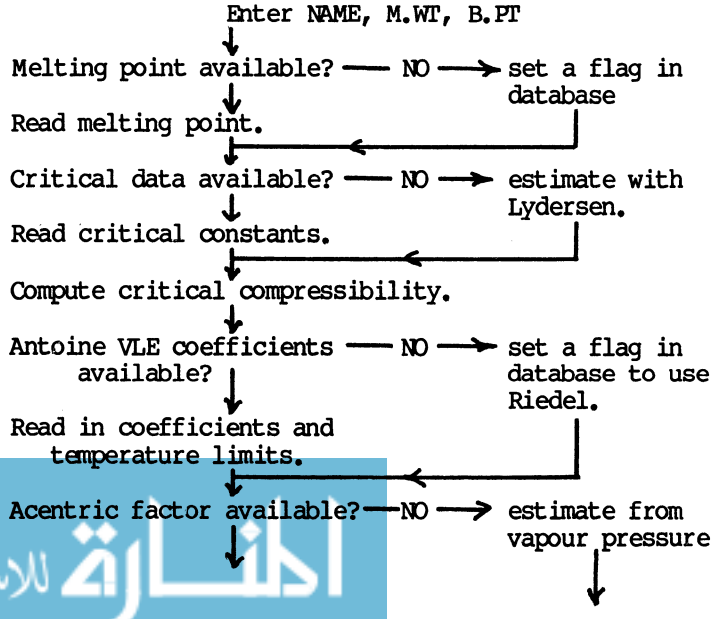
USE OF THE PHYSICAL PROPERTY DATABASE

The obvious use of the package is for obtaining point data on pure copmponents and mixtures at a specified temperature and pressure. However, it can be used effectively for other purposes. For example, the calculation of enthalpy from temperature is relatively straightforward but the reverse is not necessarily true.
If the enthalpy of a mixture is known, the database may be invoked and a temperature guessed, leading to a calculated value of enthalpy. It is usually possible to arrive at the correct temperature by trial and error in a few tries; the procedure typically takes 2-3 minutes, which is considerably faster than attempting the calculation by hand.
As another example, an adiabatic flash calculation can be performed by calculating a mixture vapour fraction at a specified temperature and pressure, and then calculating the enthalpy. By a series of trial and error calculations, the vapour fraction and final temperature of the mixture can be easily arrived at for the lower flash pressure.
The database package is used at present for research and teaching and is run on CROMEMCO microcomputers with 64K of memory. Although intended for

microcomputer use, it has been recompiled and run on a DEC PDP 11/34 and an IBM 370.

ADDING DATA TO THE DATABASE

To add data to the database, a facility is provided called DATAGEN. The structure of this package is very similar to the database package in that it consists of an executive which supervises a set of data generation routines and accesses the database. The user participates in an interactive session with the package and is prompted to supply items of data. If the data is not available, DATAGEN will estimate it. The data which is then appended to the database can be all correlated experimental data or a mixture with estimated data. In the extreme, the user need only supply a name, the molecular weight and the normal boiling point.
When DATAGEN has been run, a DATA REPORT is stored in a file and the user then has the option of having this printed. The data report is a valuable document for recording the source and accuracy of data and whether or not it is estimated. Reports are produced in a standard format for ease of filing. A further option allows the user to reject the data rather than add it to the database. This is useful where the implications of estimating properties are not known. A data report can be generated and examined and, if necessary, different options in DATAGEN can then be selected before the data is finally appended to the database.
DATAGEN is best explained by the following flow chart (Figure 5):



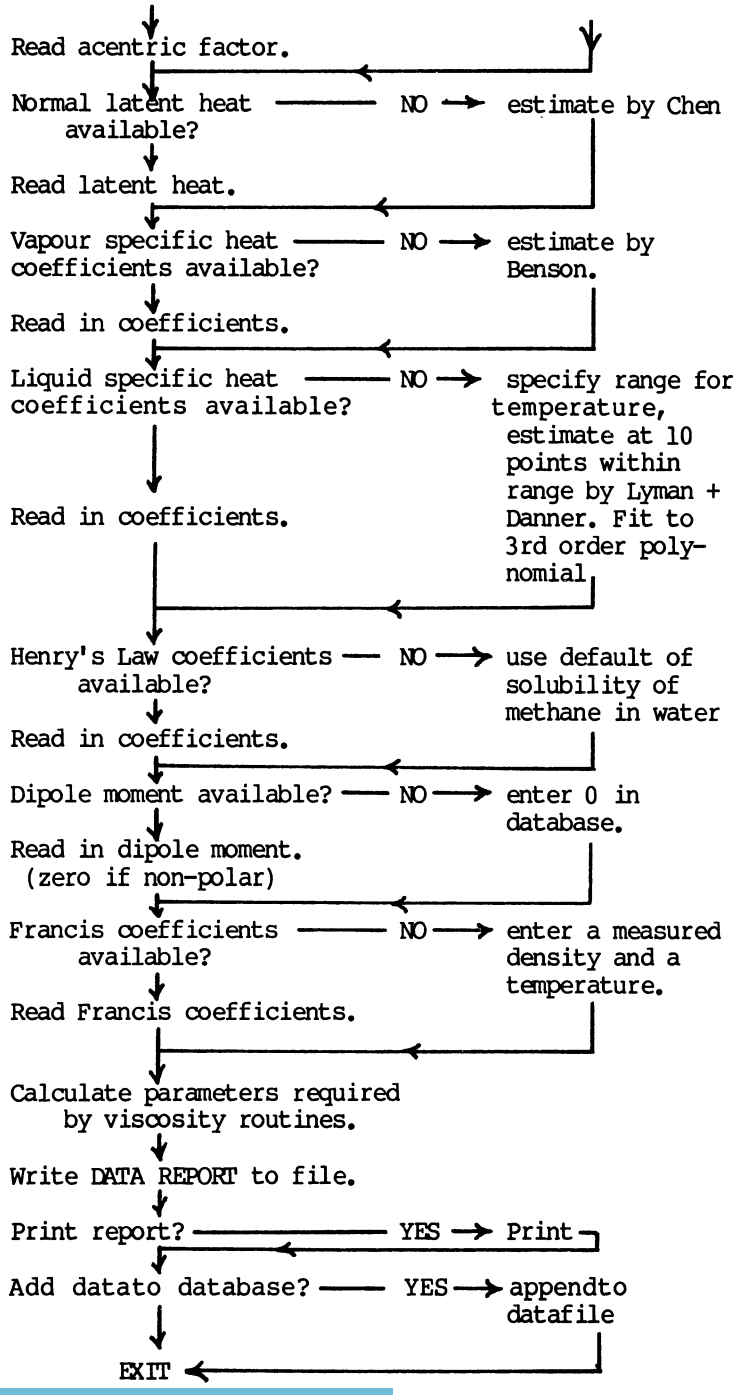


Figure 5. Flowchart for DATAGEN.

EXPANSION OF THE PACKAGE.

As stated above, there is limited room for further expansion of the package in its present form if it is to be run on a 64K, 8-bit microcomputer. The package normally runs under the CROMEMCO CDOS operating system (an extended version of CP/M) and allowing for this, there is about 50K of user space. However, after compiling the FORTRAN modules, they need to be linked and the linker requires a further 8K. The maximum compiled code size is thus about 42K. The Department also operates the CROMIX operating system on a multi-user CROMEMCO micro. Each user has a bank of 64K memory and the operating system only requires 1K of this. Furthermore, a virtual linker is available in CROMIX which does not intrude on the memory space available when linking modules. So the memory available to compiled code is 63K. While this is some improvement, it is only of limited value in extending the scope of the package.

Options for further expansion
There are three main options available.

Overlaying The time-honoured method of squeezing extra code into limited memory is to use the technique of overlaying. The programme is structured into modules so that a portion is loaded and run and when part of that is not required further it is overlaid with a new portion of the programme. This is certainly feasible but usually quite difficult to do unless the programmer has considerable experience and is fully aware of the flow of information and the location of variables. An overlay linker is available on CROMEMCO machines and probably for several other 8-bit micros. This technique is not applied to the Material Property Database.

Larger memory Small, powerful, 16-bit machines are now becoming available with the potential of addressing directly 16Mb of memory. These machines use faster microprocessors, such as the Motorola 68000, and providing suitable compilers are available, large, fast programmes can be loaded directly into memory. This may be a promising direction in which to go since the size and cost of these machines is comparable to the present 8-bit micros. The package is being revised and extended, using FORTRAN 77, to run on a MC68000 based CROMEMCO machine. However, there will still be a large number of 8-bit machines around for some time for which there may be a requirement to run packages like the one under consideration here.

Exploiting the Operating System The most promising line of development seems to be to exploit the operating system. This is the line favoured at present in the development of this package and work is continuing on the use of CROMIX. This operating system is very similar to the user to Bell Labs' UNIX. UNIX appears to be developing as the prime candidate for the standard operating system for 16-bit machines. Because of its popularity, UNIX has been implemented on a variety of machines, including microcomputers, and there are imitations under various names for both 8 and 16-bit machines. CROMIX is CROMEMCO's development and was originally written in Z80 code for their 8-bit micros; it is now also available on their MC68000 based series.
CROMIX has most of the facilities normally found in UNIX together with some additional features. One of these is the ease with which 'command files' can be written and used. The programmer writes command lines (the commands normally entered at the screen) into a file with a .CMD extension to its name and then places the file in a directory called /CMD. A user may then just name the file for it to be automatically found, read and the command lines executed. A feature of the command language is that it may be tested and 'programmed'. Figure 6 gives the listing of a typical command file.

FILE NAME: BANK.CMD

```
1.      %start
2.      echo "enter property code number"
3.      input > temp
4.      testinp temp 1 2 3 4 5 6 7 8 9
5.      if -err goto first
6.      %second
7.      BANK2
8.      goto finish
9.      %first
10.     BANK1
11.     %finish
12.     echo "do you want to continue?"
13.     input > temp
14.     testinp -d temp NO
15.     if -err goto start
16.     echo "end of run"
```

Figure 6. Command file BANK.CMD

To run an enlarged Material Property Database package, the user would enter the command BANK, but instead of the FORTRAN programme BANK being loaded, the command

file is run. The user enters the ID of the property required and this is sent to a temporary file, TEMP. The next command tests the contents of TEMP against a string of IDs (1 to 9). If a match is found, module 2 of the package is loaded. This might well be an information base. The contents of file TEMP can further be used to extract the required information from the module. If there is no match in the TESTINP command, an error signal is generated which forces the commands after the label 'first' to be executed, ie., module 1. Module 1 might be the present physical property package.
In principle, using this technique, there is no limit to the number of modules which may be called under the 'blanket' title of BANK. The user is unaware that he is not executing a single large programme. This enables a large package to be run in limited memory.
The main restriction on performance with this approach is that a lot of file handling is required. With a system based entirely on floppy discs, this would be slow, but not impossible. In practice, it would be advisable to have a fast, hard-disc system available.

Development of the package
The present physical property module will be extended to include desirable functions such as thermal conductivity, liquid viscosity, diffusivity and surface tension. Beyond that, the CROMIX (or UNIX) operating system is being exploited to enable modules to be included covering aspects such as solids handling, health and safety data [Sax, 1975], and other relevant information about materials.

References

Reid, R.C., Prausnitz, J.M., Sherwood, T.K. (1977)
The Properties of Gases and Liquids. 3rd Edn.
McGraw Hill Ltd.

Sax, N.I. (1975)
Dangerous Properties of Industrial Materials.
Van Nostrand Reinhold

A VIRTUAL STORAGE DATA MANAGEMENT SYSTEM FOR FINITE ELEMENT ANALYSIS ON MICRO'S.

E. Backx & J.P. Rammant

K.U. Leuven & SCIA S.V.

ABSTRACT

Micro- and minicomputers have limited internal memory capacity. All finite element analysis programmes therefore make extensive use of complicated overlaying techniques together with COMMON data blocks. Through this structure programme errors are difficult to recover.

An intelligent data management software is presented typically for micro's and written in BASIC, to overcome the mentionned finite element disease. It's basis is a structured data access method that stores all input arrays as text on external storage (hard disk). A small subprogramme keeps track on input, editing, additions of the input data.

Appropriate pointers and array names replace the normally dimensionned matrices. Through the analysis virtual pages of these data are loaded and shifted in the various programme modules (dynamic allocation). Also the results are stored in the pages, facilitating the postprocessing work.

In the paper a full description of the method is given together with a programme listing and examples of achieved results.

The methods used have been implemented on Wang-computers. It is not certain that it can be used without change on other desktop computers.

INTRODUCTION

Scientific programs are usually written in FORTRAN. However, because of the introduction of relatively cheap desktop computers which become more and more powerful, the BASIC language is also becomming a popular programming language among engineers. Although "STANDARD" BASIC has a limited applicability, some of the extensions proposed by manufacturers of computers, make BASIC almost as powerful as FORTRAN 77. It is even very likely that in the near future, the intrinsic drawback of BASIC, namely the reduced speed of execution because it is an interpretive language, will be removed by the use of a BASIC compiler. This compiler is already available for some computer systems. At that moment, the only remaining problem is the compatibility between systems. At present, once you have decided on a system, and you have developed some software, you cannot replace the system without rebuilding the software.

Structural computations (finite elements, displacement method) require a large amount of data that has to be used in different parts of the program. Generally, data space is allocated by the use of fixed size arrays (static allocation). In that case, depending on the structure to be calculated, there will be unused space in most of the arrays. On the other hand, if you need more space for a particular array, the program has to be changed. To overcome this problem, in FORTRAN one can use dynamic allocation by specifying a single large array and by defining pointers when calling a subroutine. These pointers indicate the first elements of the different arrays. This principle cannot be used in BASIC. However, one can use a different approach.

The different parts of a program in a desktop computer are chained. That means, that after execution the source in memory is completely or partially deleted and new source is loaded from disk. At the same time, all variables and arrays which are not in common, are deleted.

INTRODUCTION

It is possible to use common variables for specifying the size of arrays defined in the newly loaded source.
So, you can use arrays with the appropriate size. This enables a programmer to avoid empty spaces in arrays. However, if the memory available is to small to allocate space for all the arrays, the program stops and the problem cannot be solved.
This is so for every program and is not dependent on the programming language.
To remove this problem as well, it is sufficient to realise that, when the program is running, at each moment, there are only very few variables needed, because only the variables used in a particular statement have to be known.
The arrays which are not needed can be in memory, if there is sufficient space available, or can be on secondary storage.
To verify if the required data are available, a data manager is used. Once a data manager is used to allocate space in primary memory, it can also be designed to manage the data on secondary memory and it can become a data base management system if it can be used by different programs and programmers.
There are different levels of sophistication in data base management systems. The system described in this article is a very simple manager, specially designed to handle variables in arrays.

REQUIREMENTS

A data management system for structural computations on small computers should take into account the following considerations :

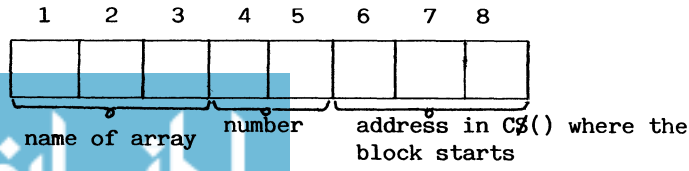
- the size of primary memory is variable
- the size of secondary space (disk) is variable
- the primary memory is relatively small (32-64K)
- the arrays have large differences in size
- the number of accesses to secondary space should be as small as possible
- avoid reordering data on the primary space and on the secondary space

REQUIREMENTS

- the code should be small in size
- the use should be very simple
- the data base on secondary memory can be accessed from different programs
- new arrays on secondary space can be created and existing arrays can be deleted
- the space used on secondary memory is always compacted.

DESCRIPTION

To avoid the necessity to reorder data, constant length blocks are used in primary and secondary storage. This length is F2 bytes. (We use variable names which are also used in the program listing. This will enhance the understanding of the code.) Arrays may be longer than one block. In that case, they are splitted into different arrays which are called by the same name. The data manager will identify the array by a name (3 letters) and a number. This number however has not to be specified by the user but is obtained from the index number. For instance, the coordinates of node 1 will be found in block CORØ1.
If a block can contain the coordinates of N1 nodes, than the coordinates of node N1+1 will be in block CORØ2.
The blocks which are available in primary space are kept in the string C$() which length is a multiple of F2. The size varies and depends on the memory size available. A dynamic allocation is used to make sure that at each moment, the complete available memory is used by the program code, the fixed sized arrays, the variables and C$().
To know where in C$() a particular block starts, a catalogue S1$(VØ)8 is used. The catalogue S1$() has VØ entries, each 8 bytes long. The content of one sequence of 8 bytes is :



DESCRIPTION

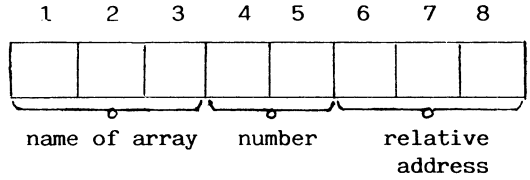
The number of entries, VØ in S1$() equals the number of blocks that can be defined in C$().

On secondary storage, a similar data structure is used. The disk is divided into sectors of 256 bytes. The choice of F2, is such that a few sectors are exactly filled with data, such that there is no empty space in the last sector of a block.

The data in a file is ordered as follows :

1. the catalogue S2$()
2. variables describing the size of the problem, the title, ...
3. data blocks.

At the beginning of the execution of a program, firstly the file is opened and the size is determined. The size of the file gives the number of blocks that can be stored and defines the size of S2$(F3)8. The content of each sequence of S2$ is as follows :



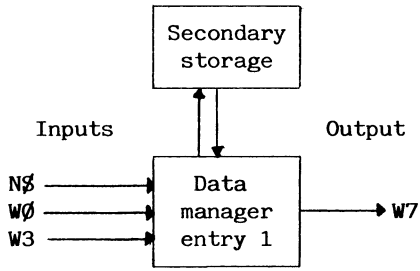
The relative address gives the number of sectors from the beginning of the file until the first sector that contains the array given in bytes 1 to 5.

At the beginning, S1$() is empty. When the program needs a variable that is stored in the data base, the data manager is used to find out if it is in primary memory. If it is not in primary memory, the search continues in secondary memory. If it is not in secondary memory, a new block is initialised. If it is in secondary memory, the block which contains the variable is loaded in primary memory.

Prior to this loading however, the program checks if there is primary space available to load the new block. If there is no space available, a block from primare space has to be saved on secondary storage. The following figure shows the block diagram of this part of the data manager.

DESCRIPTION

The flow chart of the data manager itself is shown afterwards.



Meaning of variables : N$ 3-character name of array
W$ index
W3 length of information
W7 starting address of required information in C$()

Block diagram of Data manager

The variable W3 deserves some further explanation. The information we want to handle by the data manager are sets of variables in arrays.

In finite element programs this can be :

- the coordinates of a node (x,y,z)
- the nodes defining an element (I,J,K,L)
- the parameters describing a material (E,ν,t ...)
- the parameters describing a load (code, nodenr., load, direction ...).

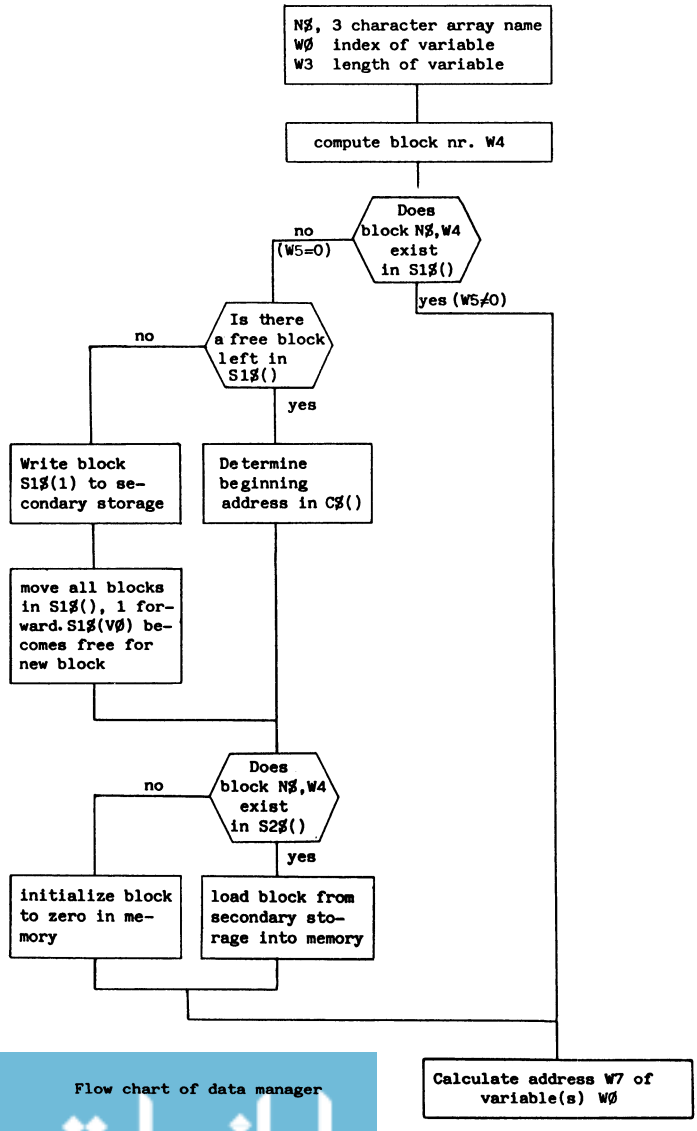
The information can consist of real numbers (i.e. coordinates), integer numbers (i.e. element definition) or can be mixed (i.e. load description). The data manager will only handle strings of bytes. Therefore, the information has to be converted to the appropriate format. Some examples follow :

- Coordinates of a point j are given by x,y,z. These coordinates will be put in the data base with image (+##.## ## ## ###↑↑↑↑), this is with 6 significant numbers. The data will be converted by a PACK statement. Each coordinate will then take 5 bytes.

DESCRIPTION

So the three coordinates together will occupy 15 bytes. (W3=15)

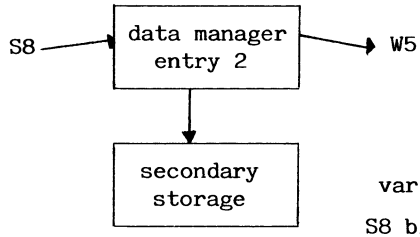
- Nodes defining an element are given by I,J,K,L. These nodes will be conserved with 4 significant numbers (maximum nodenr. allowed = 9999).
  Each node will use 2 bytes, the definition of the element will take 8 bytes. (W3=8)



Flow chart of data manager

DESCRIPTION

A second entry in the data manager is called at the end of a program module when a new module has to be chained in.
This entry is used to write the data in primary space to secondary space. The block diagram is shown here :

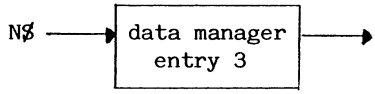


variables :

S8 block number in S1$()

W5 block number in S2$()

A third entry in the data manager is used to delete an entry from secondary storage.



variables :

N$ 3 character name of array

The data in the blocks are not deleted but the name of the array in the catalogue is deleted, so that the entry becomes available for other arrays.

IMPLEMENTATION

The listing of the data manager in BASIC for Wang Computers is given in the Appendix 2. The meaning of the variables is given in Appendix 1.
In the implementation, the use of string handling and search commands, which are available in WANG Basic, turned out to be very useful.

APPLICATION

The data manager described here was implemented in the ESA programs which are developed, commercialised and maintained by SCIA S.V., Belgium. (E. Backx 1978, J.P. Rammant 1981, 1982) The use of this data manager makes it possible to use the programs for specifications which where not possible up to now. Virtually, there are no limits to the number of nodes, elements, material properties, boundary conditions, load cases, load combinations, as long as there is sufficient space available on secondary storage to hold the data and to solve the equations. One of the largest actual cases that was run had the following specifications :

number of nodes	: 941
type of element	: truss element (3 D.O.F.)
number of elements	: 3880
number of loads	: 1004
number of load cases	: 4
number of load combinations	: 4
number of equations	: 2823
primary memory	: 56 Kb
secondary memory	: 50 Mb

The data manager makes it possible to write post-processors much more independent from the main program.
Also the selection of output to be printed is greatly simplified.

EXTENSION

The use of a file on disk which data structure is controlled by a data manager, allows a user to create input data on a small computer configuration and to run the computations using this data on a larger configuration. It is also possible to copy the data file to a diskette and to send it to a customer or to the developer for further manipulation or to correct program-errors.

EXTENSION

Because also the results of the computation are put in the data base, the file contains the complete information regarding a structure, thereby eliminating the need to print every result on paper.

The advantages of the system are further enhanced by the use of a library-file or library-disk. This library can contain several projects and is built up using similar principles as the data file. That means, there is a catalogue in which the project name, the beginning and the end of the data are conserved. The length of the different members is now variable.

It is possible to copy the data file to the library, so that a new project can be started. Data files can also be deleted from the library. From an existing library, a new library can be generated which contains the selected members.

CONCLUSIONS

A data manager for a desktop computer has been written. The programming language was BASIC. It has been implemented successfully in the ESA programs on WANG computers.

The data manager was specially designed for scientific data which is normally handled by arrays defined by static or dynamic allocation. The advantages of such a data base manager are :

- avoiding problems caused by insufficient memory
- ease of programming
- improved use of space on secondary storage
- exchange of data is possible between programs and between different firms
- in combination with a library, a very versatile tool is obtained than can greatly reduce the need for printed output.

REFERENCES

Backx E., Rammant J.P., Schymkowitz G. : "SAP Runs on a 16 K Desk Computer", Third SAP Users Conference, Los Angeles, 1978.

Rammant J.P., Backx E. : "STEELSTRAC : Steel Structures Automated Coding", IABSE Colloquium on Informatics in Structural Engineering, Bergamo, October 6-8, 1982.

Rammant J.P. : "SCIA's Finite Element System on Desktop Computers", Finite Element Systems Seminar, Southampton, September 23-25, 1981.

ACKNOWLEDGEMENTS

The author would like to thank SCIA S.V., Attenrodestraat 6, B-3385 MEENSEL-KIEZEGEM, Belgium for their autorization to publish the article.

APPENDIX

APPENDIX 1 : Variables used in the program listed in Appendix 2

C$()	array containing data in memory
C1$()	auxiliary array to load and save data to disk
F3	length of 1 block
GS	first sector of file on disk
H$	auxiliary variable
N$	string containing name of block
SØ	relative address of block on disk
S1$()	catalogue of blocks in memory
S2$()	catalogue of blocks on disk
S4	address of first variable of block in C$()
S5	number of variables in 1 block
S7,S8,S9	auxiliary variable
VØ	number of entries in S1$()
WØ	index of required variable
W3	length of variable
W4	index of first variable in block N$
W5	auxiliary variable
W7	address of required variable in C$()
W8	auxiliary variable

APPENDIX

APPENDIX 2

```
9900 DEFFN'70(N$,W0,S4,W3,W7)
    : S5=INT(F3/W3)
    : W4=INT((W0-1)/S5+1)
    : S5=(W4-1)*S5+1
    : CONVERT W4TO STR(N$,4,2),(##)
    : MAT SEARCHS1$(),=STR(N$,1,5)TO H$STEP 8
    : W5=INT((VAL(STR(H$),2)-1)/8+1)
    : IF W5=0THEN 9905
    : UNPACK(#######)STR(S1$(W5),6,3) TO S4
    : GOTO 9950
 9905 MAT SEARCHS1$(),="      "TO H$STEP 8
    : W5=VAL(STR(H$),2)
    : IF W5=0THEN 9920
 9910 W5=INT((W5-1)/8+1)
    : IF STR(S1$(W5),6,3)="    "THEN S4=(W5-1)*F3+1
    : ELSE UNPACK(#######)STR(S1$(W5),6,3)TO S4
 9915 GOTO 9935
 9920 GOSUB '73(1)
 9925 UNPACK(#######)STR(S1$(1),6,3)TO S4
    : IF V0=1THEN 9930
    : FOR W5=1TO V0-1
    : S1$(W5)=S1$(W5+1)
    : NEXT W5
    : W5=V0
 9930 W5=V0
 9935 PACK(#######)STR(S1$(W5),6,3)FROMS4
    : STR(S1$(W5),1,5)=N$
 9940 INIT(00)STR(C$(),S4,F3)
 9945 MAT SEARCHS2$(),=STR(N$,1,5)TO H$
    : W5=VAL(STR(H$),2)
    : IF W5=0THEN 9950
    : W5=INT((W5-1)/8+1)
    : UNPACK(#######)STR(S2$(W5),6,3)TO S0
    : DATA LOAD DA T#3,(S0+G5,W8)C1$()
    : STR(C$(),S4,F3)=C1$()
 9950 W7=S4+(W0-S5)*W3
    : RETURN
 9955 DEFFN'73(S8)
    : IF STR(S1$(S8),1,5)="      "THEN RETURN
    : MAT SEARCHS2$(),=STR(S1$(S8),1,5)TO H$STEP 8
    : W5=VAL(STR(H$),2)
    : IF W5=0THEN 9960
    : W5=INT((W5-1)/8+1)
    : GOTO 9965
 9960 MAT SEARCHS2$(),="      "TO H$STEP 8
    : W5=VAL(STR(H$),2)
    : W5=INT((W5-1)/8+1)
    : STR(S2$(W5),1,5)=STR(S1$(S8),1,5)
 9965 UNPACK(#######)STR(S2$(W5),6,3)TO S7
    : UNPACK(#######)STR(S1$(S8),6,3)TO S9
    : C1$()=STR(C$(),S9,F3)
    : DATA SAVE DA T#3,(S7+G5,W8)C1$()
    : RETURN
 9970 DEFFN'74(N$)
 9975 MAT SEARCHS1$(),=STR(N$,1,3)TO H$STEP 8
    : W5=VAL(STR(H$),2)
    : IF W5=0THEN 9980
    : W5=INT((W5-1)/8+1)
    : INIT(20)STR(S1$(W5),1,5)
    : GOTO 9975
 9980 MAT SEARCHS2$(),=STR(N$,1,3)TO H$STEP 8
    : W5=VAL(STR(H$),2)
    : IF W5=0THEN 9985
    : W5=INT((W5-1)/8+1)
    : INIT(20)STR(S2$(W5),1,5)
    : GOTO 9980
 9985 RETURN
```

NETWORK DATA MANAGEMENT SYSTEM
GENERAL ARCHITECTURE AND IMPLEMENTATION PRINCIPLES

W. Staniszkis (*), M. Kowalewski (*), G. Turco, K. Krajewski (*), M. Saccone
(*) On leave from CPiZI, Warsaw, Poland

CRAI (Consorzio per la Ricerca e le Applicazioni di Informatica) - Rende (Cosenza), Italy
This work has been supported by CNR-Progetto Finalizzato Trasporti

Abstract
The general presentation of the Network Data Management System is provided. The objectives and the application background of NDMS is discussed and the influence of these factors on the NDMS architecture and implementation approach is illustrated. Finally, the distributed transaction processing control and the query processor facilities are presented.

OBJECTIVES AND APPLICATION BACKGROUND

The need for distributed management and processing of data has been identified as one of the prime requirements of the National Transportation Information System. The NTIS project is currently carried out by CRAI as an applied research project sponsored by the National Research Council (CNR) of Italy. The main objectives of NTIS are the following:

1) to support the optimal employment of resources (energy, machinery, manpower).
2) to provide sufficient information pertinent to the strategical,tactical planning with respect to the transportation services offered (including their quality and security). This entails such elements as the policy regarding taxes, loans and tariffs and the traffic control activities.

Such objectives may be supported only if the appropriate information is made available to the respective authorities, considered to be the prime users of the NTIS. We have identified the following categories of data that is to be maintained in the system:

1) the statistical data collected on a periodical

basis
2) the management data
3) the traffic control data representing the dynamic features of the national transportation system.

The high level of availability is considered mandatory in the case of the latter two categories of data. This will include the data regarding personnel, the state and planned extensions of the transportation network, transportation media and principal carriers, cargo and passenger flow as well as the economic parameters of the system. Most of the required data is already available in the information systems that are in operation in the various ministries involved in transportation problems such as the Ministry of Industry etc. as well as in the regional administration authorities. The main problem results from the fact that these data resources are poorly coordinated and the level of availability of the required data is very low.

There are two possible solutions of the above problem. One is to create a centralized data base system supported by data extracted from all of the relevant information systems.

In such case the data base may represent an arbitrary level of abstraction with respect to the national transportation system. This may be appropriate for such categories as statistical data and the economic parameters but it will not be sufficient for management and traffic control data.

The other approach is to make the existing data accessible to the NTIS users. This may be achieved if the required integration of data is supported to allow for a standardized view of all of the diverse representations of data, as well as for the flow of data between the underlying information systems. Such a solution may not interfere in a significant way with the information systems that are presently in operation.

The latter of the two possible solutions has been chosen and the Network Data Management System was designed and is at present under development to become the principal software tool for design and implementation of the NTIS. Similar approach has been adopted by others and such systems as Multibase (Landers 1982), R* (Daniels 1982), POREL (Neuhold 1982) and Sirius Delta (Litwin 1982) are being developed. Similar problems have been also identified in SDD1 (Bernstein 1981), although in this case we deal with one distributed data base, not with a collection of geographically distributed data bases as in the case of the NTIS.

The requirements and design principles for distri-

buted data management systems were defined in (Bernstein 1981, Davenport 1981, Davenport 1978, Levin 1975).
Problems related to the integrated view of distributed data in the heterogeneous DBMS and hardware environment were discussed in (Anderson 1979).
The following features have been considered as the prime design and implementation of NDMS:

1) the system is to facilitate the integration of data residing in geographically distant computer systems and manage diverse data base management and/or file management systems.
2) The integration of diverse information systems is to be achieved by providing for the exchange of messages and data between application programs.
3) The global view of data is relational and query processing is supported by a query language defined as a subset of SEQUEL 2 (Szymanski 1980). The geographical distribution of data is to be transparent to the end user and all of the necessary mappings of data are to be supported by NDMS.
4) The standard network control software (like SNA, DECNET etc) is to be used for the computer network control.
5) The system functions should be completely transparent to the local information systems.

The general NDMS application environment is shown in

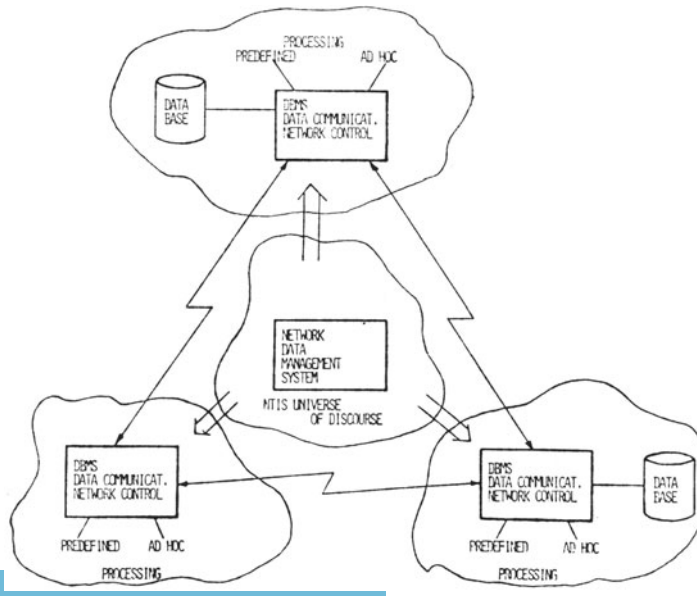


Fig.1. NDMS application environment.

transactions must be rolled back.

- The data exchange function that allows for sending data from the source to the destination point, which are tasks of various types, residing in various nodes, already waiting for data, or initialized dynamically. This function includes usage of standard communication features (network control system) and the NDMS data protocol.
- The System Encyclopaedia contains all the information pertinent to the distributed environment. The system is similar to the typical Data Dictionary/Directory, however some differences should be noted.
  First, a DD/D part of the S.E. is related to the global view of data integrated with information about geographical placement of data, as well as the quantitative characteristics of the local data bases. It contains also the network definition and the application program (transaction) definitions.
- The global back-up and recovery. It is the responsibility of the local DBMS to provide this function for the data base they manage. In the distributed environment the global consistency of data is necessary. This problem is to be handled by the application design and implementation (it is concerned with update transaction only), nevertheless some additional mechanisms within the NDMS should be provided.
- The support functions and utilities. This is a set of on-line and off-line procedures allowing for generation of all necessary definitions, mapping rules, control tables, etc., as well as some auxiliary functions as for example file transfer, operator's aid and so on.
- The performance monitoring. During run time of the NDMS some quantitative information about the NDMS activities should be collected, especially those concerned with the data exchange. This statistical information may be then subject to further analysis in order to estimate the global NDMS performance. Results of such an analysis may cause modifications of some quantitative data descriptions (having influence on queries optimization), increase of hardware capacity, system's and application's development policies, etc.

The schematic representation of the NDMS architecture is shown in fig. 2.

The central part of the system is the System Encyclopaedia comprising the entire definition of the NDMS application environment. The data definition contains the global relational schema, together with

fig.1. All of the predefined processing functions are to be designed and implemented as on-line transaction processing applications with the application programs executed in the various computer network nodes exchanging messages and data. Ad hoc processing is supported by SEQUEL2 defined as the NDMS query language and all of the queries are to be defined against the global relational view of data.

ARCHITECTURE OF THE NETWORK DATA MANAGEMENT SYSTEM

The architecture of NDMS discussed in this sections reflects the NDMS features as implemented on each node of the computer network. However, holding the global view of the distributed system, the NDMS users (such as application programmers, end users and the network administrator) deal with only one integrated network (called the NDMS network) regardless of the various network control systems that may be used. The NDMS network may be characterized by the following features:

1) the NDMS data protocol is employed for node to node communication
2) the terminal points of network messages are NDMS software routines residing in all nodes of the computer network
3) the message routing, control and recovery is managed by the standard network control software
4) the internetwork gateways (e.g. SNA-DECNET gateway) are used to provide for the integration of the NDMS network.

The above NDMS network characteristics are matched with the NDMS software features summarized below:

- the Global Data View, is the specification of the relational structure of data, that is stored in one/many/all node(s) and that may be accessed from any node. One can say that the global data view is a kind of relational super-schema.
- The Query Facility is a query language processor that allows the end user to formulate a query in a Sequel-2 like language and to obtain required data using one of the user views defined upon the global view.
- The Transaction Processor. This facility is similar to other transaction processing systems, however it enables execution of transaction oriented towards the distributed environment.
- The logical transaction program coupling providing for the concurrent processing of a number of transactions in different nodes. Results of such processing depend on completion of all coupled transactions. If one fails, all the

all of the user views defined upon this schema. Materialization of the global relations (those defined in the global relationed schema) is controlled by the global mapping rules, providing information about the geographical distribution of data and the global relation derivation rules (i.e. a global relation may be defined as a union of local relations of the same type). Materialization of the local relations is controlled by the local mapping rules providing all the necessary information for data model transformation (e.g. from the Codasyl-type network data model into the relational data model). All of the quantitative data characteristics required by the query optimizer are also stored with the data definitions.

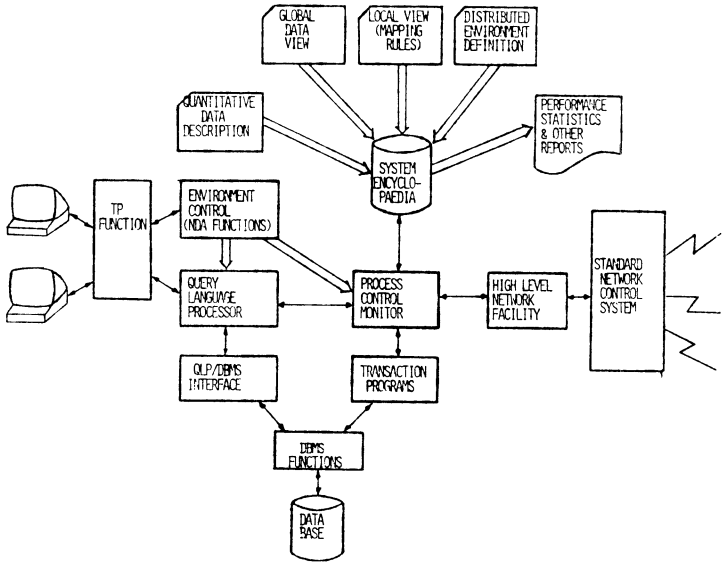


Fig.2. NDMS Architecture

The NDMS network components are also defined in the System Encyclopaedia comprising such elements as the network node description, definition of application programs (transactions) and system's users together with the security and authority rules.
All of the information stored in the System Encyclopaedia may be viewed as the relational data structure (regardless of the geographical distribution) and it is, therefore, accessible via the NDMS query language.
The Process Control Monitor initiates and controls execution of all of the processes that may be either application processes (transactions or queries) or

system processes. It is also responsible for the maintenance of the System Encyclopaedia.
The Process Control Monitor executes all of the NDMS auxiliary functions such as the NDA utilities, the performance monitor etc.
The Query Language Processor parses the end user queries or accepts partial queries that may arrive from other NDMS nodes, decomposes the end user queries into a number of partial queries that are to be executed in other nodes and provides for the overall query optimization. The QLP module residing in the query originating node controls the execution of the query materialization strategy.
The QLP/DBMS Interface provides for the transformation of the data model of the local DBMS or the File Handler and materializes the local relational model according to the local mapping rules defined for each NDMS node. Such a module has to be implemented for each DBMS type involved in the NDMS application.
The High Level Network Facility constitutes an interface between the NDMS software and the respective network control system. It allows for implementation of the standard NDMS network protocol regardless of the network control system used at any particular NDMS node.
The following types of processes may be executed in the NDMS environment:

- Ad hoc queries handled by the Query Language Processor
- Transactions retrieving and/or updating data distributed over many nodes. Transactions may fall into two categories:
  . queued transactions which are sent from node to node, then stored in the transaction queue to be later processed depending on some events, usually as soon as possible
  . logically coupled transactions, organized in such a way that a transaction invokes a set of other transactions to be processed in specified nodes. All these transactions run simultaneously and depending on processing results they complete processing normally or restore previous status of all data bases.
- On-line utilities and support functions, that are dedicated to end users and network data administrator
- Internal NDMS processes.

The users of the NDMS fall into three categories, namely the end users that are considered to have no data processing background, the application programmers and the network administrator. Different system facilities are oriented towards these diverse

user groups, according to their needs and qualifications.
The end users are provided with the following mechanisms:

- The Query Language Processor, executing ad hoc queries formulated in SEQUEL2.
- Predefined transaction. For the end user there is no difference between the conventional transaction processing and the NDMS transactions. Data accesses in many nodes are encoded within a transaction program.
- End user aids. These routines allow the end user to obtain information about data resources, names to be used in queries (relations, attributes) etc. Definition of such information is the responsibility of the NDA.
- Utilities. This set of functions is partially dedicated to the end user and includes among others:
  . extraction of data from data base(s) and storing them as a file for further processing (i.e. APL files).
  . Transfering data to/from other nodes.

Application programmers are to implement a collection of application programs to be executed in the NDMS distributed environment.
To allow this a set of NDMS commands is proposed:

- Bind node(s), which initializes some control blocks used in further processing
- send transaction data (initial record, subsequent data records)
- Receive transaction data
- Unbind (end of processing)

Using these commands, a transaction program which interacts with the end user is able to generate initial transaction records which include parameters and names of other transactions to be processed in specified node(s). If logically coupled processing is used, all processes , invoking and invoked, should be synchronized in terms of data exchange (task-to-task dialogue).
The Network Data Administrator is responsible for the definition of the distributed environment, including the operational and data resources. All such information is stored in a system encyclopaedia. The NDMS will provide NDA with appropriate interfaces and routines for performing this job.
During run time of the NDMS a very important function is a set of procedures (master terminal function). This will include obtaining information about the current status of the NDMS, changing NDMS parameters and others (including system close down). Since statistical information will be gathered

during system run time, providing the NDA with legible form of these statistics is another function of the NDMS. Such data may be helpful in improving the system performance, which may need some change in the system encyclopaedia (with respect to quantitative data description).

THE NDMS IMPLEMENTATION APPROACH

The implementation approach has been determined by the fact that the NDMS software is to be implemented in several diverse software/hardware environments. This mandatory feature of the system has an important impact on the cost of its implementation. It may also be necessary to provide for an orderly growth of the system by allowing new software (such as a new DBMS) and hardware environments to be added to the NDMS network. In order to accomodate the above requirements with the minimum implementation cost we consider the system portability to be one of the major design objectives. Consequently, all of the software environment independent logic has been identified and the respective NDMS modules constitute the kernel of the system. NDMS kernel is being implemented in Pascal to ensure the maximum portability of the software. A substantial part of the NDMS software including the Query Language Processor, the Process Control Monitor and the high Level Network Facility is implemented as the NDMS

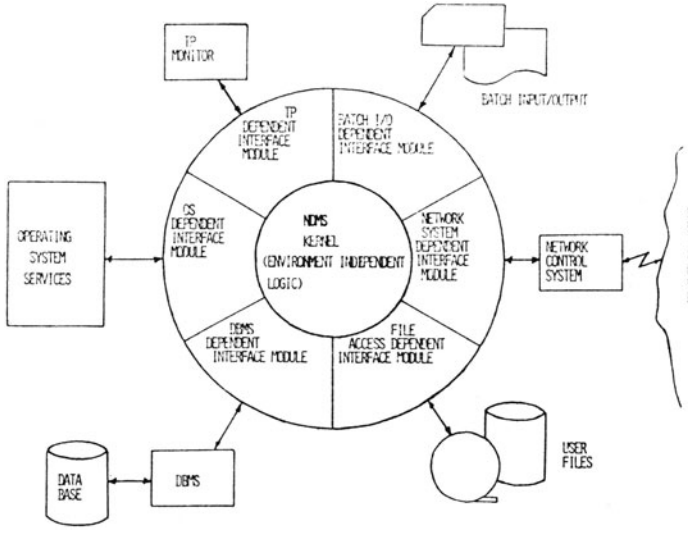


Fig. 3. The common logic approach to NDMS implementation

kernel. All of the software environment dependent features have been isolated and will be implemented for a particular environment.
In most cases (i.e. Codasyl data base management systems) we deal with the common design logic and the migration of the respective interface modules should be a straightforward programming task.
This implementation approach has been summarized in fig. 3
However such an approach made it necessary to formulate a number of specific requirements with respect to the software environment. By the software environment we mean the operating systems with access methods, programming languages, etc., the teleprocessing systems including the network control functions, the data base management systems and other specialized software components such as job entry systems, emulator programs, etc. We assume that hardware environment is of secondary importance to us, except of such obvious features as hardware capacity and the performance characteristics, since, generally speaking, the compatibility of software implies the compatibility of hardware. However,it is necessary that each candidate NDMS node is capable to maintain a physical connection with other nodes and the appropriate software support is provided. If such is not the case, the lack of the standard software support may be overcome by enhancement of the standard NDMS capabilities. In the case of diverse network control systems inter-network "gateways" should be applied. All nodes should have a multitasking feature, that is, they should be able to attach subtasks or create processes dynamically. All control components, namely data base management systems, teleprocessing monitors and network control systems, should allow for implementation of effective interface modules, that is, all management (service) functions should be supported on assembly language level. In the case of a DBMS all data manipulation commands must be supported by means of execute (reentrant) form of macroinstructions or procedure calls.

THE DISTRIBUTED TRANSACTION PROCESSING CONTROL

We mentioned above that the integration of the local information processing systems is effected by mechanisms that allow for the distributed transaction processing. The logic and the processing rules of such transactions are clearly application dependent and they are to be determined during the distributed information system design and implementation phases.

The following modes of the distributed transactions processing are available to NDMS users:

- the master transaction, to be invoked on the end user (terminal operator) request, is entered into the system via the end user interface module. The master transaction may act on the local data base just like other non-NDMS transaction programs, but it may also be involved in processing distributed data by means of other subordinated (secondary) processes invoked in remote nodes, according to requests issued by itself. There are two kinds of subordinated processes:queued and logically coupled transactions.
- the queued transaction. This transaction is invoked by the PCM according to the transaction message sent from another node by a remote master transaction. The transaction message contains the name of the transaction program to be executed as well as the processing parameters. No synchronization is assumed between primary and secondary processes, so that transaction messages may be saved in a transaction queue.

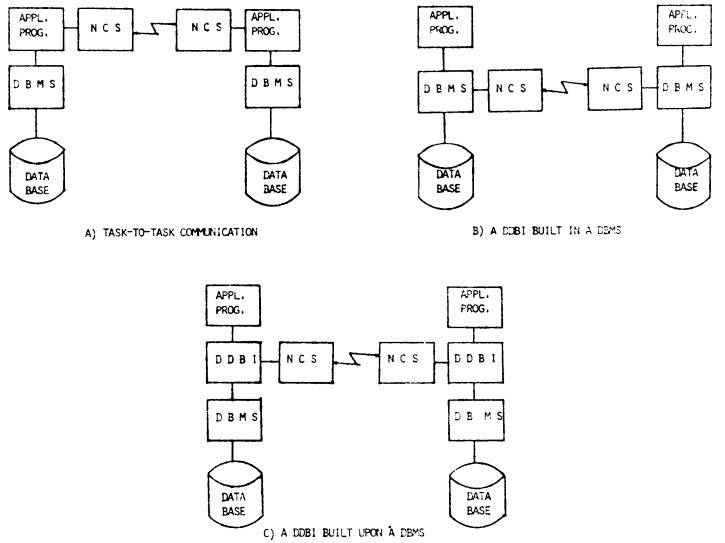


Fig.4. Methods of access to distributed data bases

- The logically coupled transactions. In such case the subordinated (secondary) processes must be fully synchronized with the master (primary) process, which is laid upon responsibility of the programmer. As for the queued transaction, the transaction message contains the transaction name and the set of initial processing parameters, but

rather than being stored in the queue, it is immediately submitted for execution. Obviously some additional data exchange (a kind of dialogue) between coupled processes is assumed. This mode of processing may be necessary whenever a number of subordinated processes should be performed concurrently in various nodes and the global data consistency should be enforced. The master transaction performs the necessary control functions, that is, if one of the secondary processes fails, the master transaction notifies the other ones, in order to initiate the transaction abort and recovery operations.

The inter-transaction communication may be implemented on different levels. Several possible implementation approaches are shown in fig. 4. One extreme solution is to provide for the direct communication (shown in fig. 4 A) between transactions executing in the remote computer network nodes. Such solution is typically used in the computer network control systems. On the other hand it is possible to implement the distributed transaction processing functions directly in the distributed data management system. Such solution is only possible if the homogeneous DBMS environment is installed on all nodes of the computer network. Since this is clearly not our case, we have opted for the intermediate solution (shown in fig.4 C). The distributed data base interface (DDBI) implemented by the High Level Network Facility and the Process Control Monitor of the NDMS allows for the necessary level of the distributed transaction processing control.

THE NDMS QUERY FACILITY

The relational data structure has been considered to be most appropriate as the basis for implementation of the NDMS query facility. A subset of the SEQUEL-2 (Chamberlin 1976) relational query language has been selected for implementation in the NDMS environment. We have included all of the selection and the view definition facilities of SEQUEL, leaving aside all of the update mechanisms originally proposed in this query language. The complete specification of the selected SEQUEL-2 subset may be found in (Szymanski 1980, RODAN 1982).

The NDMS query language may be used either in the on-line mode or in batch as the report writer. Hence, the original language has been extended with powerful data edit and report formatting facilities. Since many diverse data models (network, hierarchical, relational) may be encountered in the NDMS

environment, all of the required data mappings are supported in NDMS. The approach to data mappings in the NDMS environment is shown in figure 5. There are two distinct data mapping levels supported; the local mapping from the source data base structure into the relational data structure and the global mapping from the local relational data structure into the global relational data structure. The global relational data structure constitutes the basis for all of the user view definitions and consequently all of the user queries. In this way the complete transparency of the data distribution has been achieved. The local data mappings are supported by the mapping language definitions to be created for each source data base participating in the NDMS network. The local relation materializations and the appropriate subquery optimization are handled by the QLP/DBMS interface.

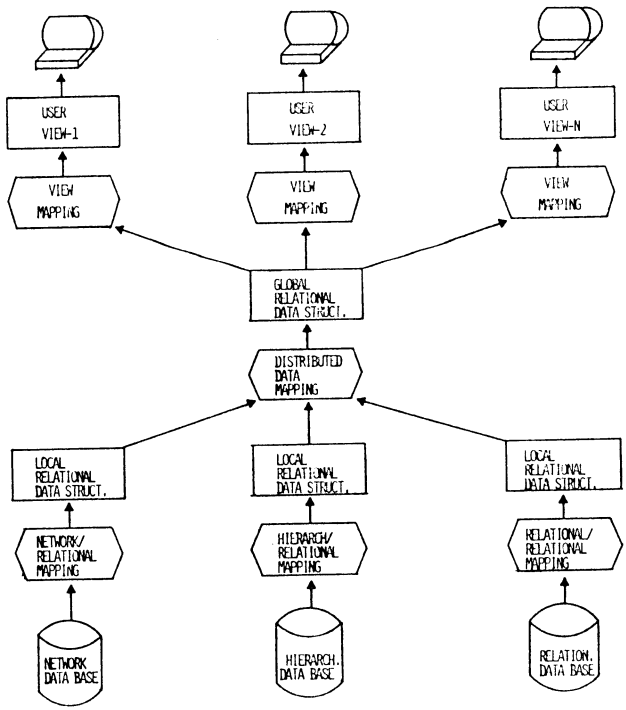


Fig.5.NDMS data mappings

The global relational queries (the ones to be executed against the global relational data structure) are decomposed into a number of subqueries to be executed concurrently in the NDMS network nodes. The query decomposition and the

resulting execution strategy is subject to the global optimization process and a strategy providing for the minimal data transmission cost is chosen.
The global optimization requires estimation of cardinalities of all source, intermediate and target relations. Such an estimation is based on the quantitative description of all of the source data bases providing data for the probabilistic evaluation of query predicates. All of the required quantitative data decriptions are to be automatically derived from the source data bases by a statistical evaluator. The statistical evaluator is considered to be a utility process to be invoked by the Network Data Administrator. The query performance parameters estimated by the query optimizer may be confronted with the actual parameters derived from the performance monitoring data and, in the case of substantial discrepancies, the quantitative data description may be corrected. The repetitive queries may be stored in the user catalogue, that may be created for each user in the system encyclopaedia, and they may be invoked by the user whenever needed. The query results (relations) may be stored as temporary relations to be available for further processing. They also may be transformed into the special format files to be used by the specialized software such as APL, econometric packages, etc.
We believe that the NDMS query facility provides a sufficiently powerful interface for most of the NDMS users.

CONCLUSIONS

Our approach to NDMS design and implementation has been biased towards the requirements of the National Transportation Information System. However, we believe that the general properties of the NDMS will make it possible to apply this tool to other distributed data processing applications. Unfortunately, the scope of this paper has not allowed for the more complete presentation of the distributed transaction control algorithms and the query decomposition and optimization. They will be presented in the forthcoming publications.

Acknowledgement

We are grateful to A. Grano and P.Rullo for their work on the query optimization and the quantitative data description. We are also indebted to W.Kaminski and St. Mezyk, the authors of the RODAN SEQUEL implementation, for providing us with many useful comments.

R E F E R E N C E S

Anderson M., Staniszkis W., (1979) System Development Facility- Long Range Design Report. Logica Ltd., London. Internal Report

Bernstein, P.A. et. al. (1981) Query Processing in a System for Distributed Databases(SDD-1) ACM Transactions on Database Systems,Vol.6,N.4,Dec.1981.

Blasgen, M.W. et. al.(1981) System R:An Architectural Overview, IBM System Journal, Vol.2O,-n.1,1981

Chamberlain,D.D. et. al.(1986) SEQUEL 2:A Unified Approach to Data Definition Manipulation and Control. IBM Journal of Research and Development, Nov. 1976.

Daniels, D. et. al. (1982) An Introduction to Distributed Query Compilation in R*. Proceedings of the 2nd International Symposium on Distributed Data Bases, Berlin F.R.G. September 1-3,1982. Edited by W.J.Schneider, North Holland Publishing Company 1982.

Date,C.J. (1981) An Introduction to Database Systems, Addison-Wesley, Reading, Mass. 1981

Davenport,R.A. (1978) Distributed or Centralized Data Base,The Computer Journal, Vol.21, N.1, 1978.

Davenport, R.A. (1981) Design of Distributed Database Systems,The Computer Journal, Vol.24, N.1, 1981.

INRIA, CNR, Bremen University (1982) Predevelopment Study on Distributed Data Management Systems, Internal Report, INRIA, France 1982.

Landers, T. Rosenberg R.L.(1982) An Overview of MultiBase. Proc. of the 2nd International Symposium on Distributed Data Bases, Berlin, 1982.

Levin, K.D., Morgan, M.L. (1975) Optimizing Distributed Data Bases - A Framework for Research. AFIPS 75, Vol. 44.

Litwin, W. et al. (1982) SIRIUS. System for Distributed Data Management. Proc. of the 2nd Internatinal Symposium on Distributed Data Bases, Berlin 1982.

Neuhold, E.J., Walker, B. (1982) An Overview of the Architecture of the Distributed Data Base System "POREL". Proc. of the 2nd International Symposium on Distributed Data Bases, Berlin 1982.

RODAN (1982) DBMS RODAN - User Manuals, CPIZI, Warsaw, Poland.

Sandberg, G. (1981) A Primer on Relational Data Base Concepts, IBM Systems Journal, Vol. 2O, N.1, 1981.

Szymanski, A. (1980) SEQUEL - A Query Language for the DBMS RODAN. Proceedings of the International Seminar on Architecture of Data Base Management Systems, Nov. 1980, Zaborow, Poland.

AN INTEGRATED APPROACH TO LABORATORY DATA MANAGEMENT, PROCESSING AND DISPLAY

G.P. Ellis

Department of Engineering, Simon Engineering Laboratories, The University of Manchester, Manchester, England

1. INTRODUCTION

A major problem in many laboratories is how to deal effectively with large amounts of data from an experimental rig, and then present the results in a useful graphical form.

This paper describes a computer-based data acquisition and processing system developed for a marine technology research project (1) concerned with hydraulic wave/current interactions and sediment transport. This experimental work produces large amounts of data (typically 1.5 million points per test), and this has necessitated the design of special acquisition software, including a novel approach to the ensemble averaging of wave data. A wide variety of data processing and graphical display techniques are employed in the analysis of the experimental data. These facilities are provided by specially written software, integrated by means of a comprehensive data management system. The resulting computer system has proved an invaluable research tool; providing rapid data collection and analysis, and great flexibility in the graphical output, whilst remaining easy to use.

2. EXPERIMENTAL EQUIPMENT AND COMPUTER HARDWARE

The experimental work is carried out in a wave/current flume of dimensions 0.5mx0.5mx22m. The waves are generated by an electromechanical piston type wavemaker with a microcomputer based signal generator (2). High quality water velocity measurements are taken using a DISA 2-channel laser

doppler anemometer (LDA) with frequency trackers. Other transducers are used to measure water heights, sediment concentrations and water pressures. A PLESSEY 1VCL computer is used for the data acquisition and processing with graphical output on a TEKTRONIX 4014 high resolution graphics terminal and a 4662 digital plotter. The computer is based on a DEC LSI 11/23 processor + 128 Kb memory, programmable realtime clock (DT2769), 16 channel ADC (DT2762), parallel and 4 serial line interfaces. File storage consists of a 5Mb fixed disc and 5Mb removeable cartridge discs; backup storage is available on magnetic tape.

3. OVERVIEW OF SOFTWARE

The majority of the computer software is written in FORTRAN IV with time critical and system orientated software written in MACRO assembler. All data from the experiments, including the results from subsequent data analysis, are stored as separate files on the removeable cartridge disc. The filemanagement system has been developed to organise the experimental data in an efficient manner and thus allow the programmer to write good, user orientated software. The file management system utilises a file directory which contains details of all the data files and this is accessed, using special routines, whenever the storage or retrieval of data is required. Comprehensive administration of this directory is also provided; for example to list or delete data files. Filenames are given in a standard format and information concerning each data file is stored in a defined manner.

This paper describes the structure of the filemanagement system and how it is implemented. The main features of the data acquisition and storage are demonstrated and examples of data processing and graphical output are given with emphasis on showing the benefits of efficient data management in the laboratory.

4. STORAGE OF DATA BY THE FILE MANAGEMENT SYSTEM

Data is stored as direct access files in RT-11 format, with a defined record size and number of records. This makes efficient use of the disc space and allows rapid access to any particular record within the file.

4.1 Filename format

The standard RT-11 filename consists of a 3 character device code (eg. RK2 implies disc unit 2); up to a 6 character name; followed by an extension code of up to 3 characters. The latter is normally used to name different types of files, eg. FOR implies Fortran source code, OBJ implies object code An example of a RT-11 filename is RK2: PROGØI.FOR

The filemanagement system utilises this filename structure by designating fields, and defining certain codes for these fields. The codes are chosen to be easily recognised.

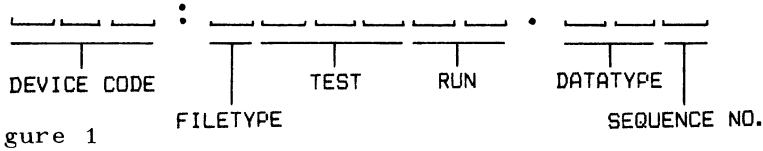


Figure 1

Figure 1 shows the standard filename format used in the file management system. A brief description of the fields and their use is given below.

(i) Device code: Normally the cartridge disc (RK) followed by the disc unit number; this is changed by the FMS depending on the availability of storage space.

(ii)File-type: Single character code used to denote the type of data stored on the file. eg. 'D'= original data, 'F' = ensemble averaged date.

(iii) Test: The number assigned to a particular test or set of experimental data. Range - 0 to 999

(iv) Run : The number given to a particular experiment during a Test. Range - 0 to 99

(v) Datatype: A two character code used to denote the source of a particular set of data.
eg. L1 - laser channel 1, W2 - wave gauge no.2
V1 - horizontal velocity, SØ - shear stress
SM - amplitude summary

(vi) Sequence number: Used when several versions of a particular file are required. For instance, when Summarising different groups of Runs from the same test.

Every Filetype has a standard file format; this facilitates recognition of the exact nature of the data from the filename and, more importantly allows automatic filename generation without reference to 'look up' tables. It also permits easy administration of existing files, such as listing or deletion, using the fields within the filename.

4.2 Header blocks

One important part of the management of data files is the ability to access information concerning a specific set of data; for example, transducer calibrations, positional information and test conditions. It is extremely time consuming and error prone for the user to enter this information everytime a set of data is processed. Therefore, in this system, blocks of information relevent to the particular Filetype are attached to the data files. These are called 'header blocks' and have a standard format depending on the Filetype. Examples of header block formats are shown in Figure 3. An interactive method of entering the initial information into the header block is used, as demonstrated in section 6.1, Date acquisition and Storage. Apart from useful information on calibration constants, Test and Run numbers, position etc., spaces in the header block are reserved for results from the data processing. For Filetype D the mean, minimum, maximum and standard deviation can be stored; this has several uses. Firstly, when a time series plot is required on the VDU or plotter the data limits are known and the axes can be scaled automatically. Secondly, if a summary of results is required from many Runs then this information can be read directly from the header block without the need to recalculate. With perhaps 32000 data points per channel in each Run this can save much unnecessary calculation time.

Figure 3 shows that spaces have been allocated for sets of results (ie mean 1 & mean 2). This is for the storage of horizontal and vertical velocities and turbulence intensities resulting from the processing of the Laser doppler anemometer data. Header block spaces marked with an "*" are used by various programs to pass information to some subroutines. Other spaces are reserved for future expansion.

Figure 3 also shows the format of the ensembled data header block, Filetype E. This information allows summaries of wave and phase data to be rapidly produced for a Test summary printout or for subsequent output in graphical form. An example of this is given later in Section 7.

5. IMPLEMENTATION OF THE FILE MANAGEMENT SYSTEM

The previous section has described the filename format used by the filemanagement system and the use of a header block to provide information about the data file. This section describes the structure of the file management directory and how it is used.

5.1 Directory structure

Experimental data and results are stored on 5 MB removeable cartridge discs, with the facility to transfer to magnetic tape for permanent storage. The RT-11 operating system assigns a device code to each side of the removeable disc, namely RK2 and RK3. When initialising a new disc a file management directory is created on device RK2. (This is independent of the operating system file directory). The management directory can hold up to 1000 file entries, each with the format shown in Fig 2.

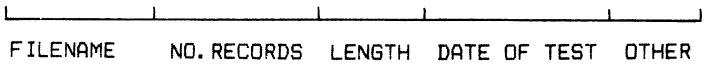


Figure 2. Format of the file management directory.

The filename is stored as a 14 character string. The number of records and record length are required by the RT-11 operating system to access the particular file. The date of the Test is stored as a code and the last two bytes are used for additional information; for example to indicate whether the data file has been written to magnetic tape. As shown in the description of the filename format, much information about the source of the data is actually contained within the filename.

5.2 Generating and checking file entries

Subroutines have been written to generate a filename in the standard format and to check whether a particular filename exists in the directory. These routines can easily be incorporated into any program which creates or retrieves data files. With subroutine EXIST the Program supplies the filename as a character array and the routine scans the management directory. If the name already exists, then sufficient information is returned to allow the program to access the file. Several techniques are used to speed up the comparison of filenames and if there is sufficient computer memory, the directory can be stored in the memory whilst running a program.

Another subroutine, FNAME, generates the standard filename given the relevant information contained in the integer header block (See Figure 3) If a new entry to the directory is required (an option with FNAME) the routine EXIST is called; this avoids duplication of filenames. Having proved a filename to be unique the routine FNAME checks that there is physical space on the first disc unit to store the particular file. If no space is large enough (RT-11 stores files in contiguous blocks) then the second disc unit is checked for space. The routine enters the appropriate device code in the filename, updates the management directory and returns to the calling program with the filename under which to store the data.

The use of these routines makes the storage and retrieval of data files automatic, as far as the user is concerned. The programmer can utilise these routines to create efficient flexible software, examples of which are shown later. Other useful routines have been written for interrogating the management directory, for example, informing the user of the next available Test or next free Run number in a given Test. The next section describes some of the programs written to administer the file management directory.

5.3 Administration of the management directory

The file management directory keeps a record of all the data files stored on a cartridge disc during the experimental work. An important requirement of a file management system is to look at the contents of the directory, delete and rename entries. The program written for this system has very flexible user input and output facilities; important features when dealing with as many as 1000 entries.

In the list, delete and rename programs the user is prompted to specify the names of the file or files. Because a standard file format is used, wildcards, in the form of '*'s can be input in any positions to indicate that all the choices in these files should be included in the file specification. See Fig. 1. Consecutive *s can be replaced by a single * and specifying the device code is optional. Example file specifications are given below.

```
RK2:E*.*  - All ensemble averaged data files
            on device RK2
*12702.*    All files for Test 127 and Run 2
*.SM        All wave summary files
```

The file specification is deciphered and a list of the selected files passed to the appropriate routine. Ambiguous or erroneous specifications are reported back to the user.

The directory listing program has options for full or brief information formats, screen or printed output and the choice of an alphanumeric sort. An example listing is shown in Figure 4. The delete file program has the option to query each selected file as a precaution against accidental erasure, and both the delete and rename programs operate on the RT11 device directory as well as the file management directory; this is necessary to keep continuity between the physical file storage and the management system.

6. DATA ACQUISITION AND STORAGE

The data acquisition system has been designed to collect data from a maximum of 16 analogue data inputs with up to 32000 samples per channel. The nature of the experiments means that the sampling has to be continuous at a rate of at least 200Hz from all 16 channels. The system can infact sample 3 channels at 1200 Hz and all 16 channels at 400 Hz; with over $\frac{1}{2}$ million data points collected in one experiment. Real time assembly language routines have been written for this purpose; the main features of these routines and their associated programs are given below, for steady flow sampling and the special case of ensemble averaged sampling.

6.1 Steady flow sampling

In this case specified input channels are sampled at a given rate, as required by experiments under steady flow conditions. First time through the data acquisition program, the number of input channels and start channel is requested together with the sampling rate and number of samples. The user starts the sampling when ready and the program passes information to the acquisition routine. The programmable real time clock is used on an interrupt basis to control the sampling via a 12 bit analogue to digital converter,(ADC), whilst

the main body of the routine deals with buffering the data and the DMA transfers to the disc store. The ADC uses sequential scanning of the input channels which can often cause phase errors between different inputs. To overcome this problem a 16 channel sample and hold board was designed and built, which effectively gives simultaneous sampling of all inputs.

At the end of the sampling period the data is held on a temporary file. The program requests the data and the Test number and, by interrogating the file management directory, can inform the user of the next Run in the specified Test. The program then gives an interactive display of header data; an example is shown in Figure 5. Information for the specified channels is retrieved from a file and displayed. Any number or code can be changed using the graphic cursor as a pointer and data channels not required for permanent storage can be thus indicated. The updated header information is stored back on file. This flexible method of entering or changing information has proved most successful, avoiding much unnecessary input of experimental constants. The program then calculates other header data and updates the file management directory. The temporary data file is split up into individual files, one for each chosen Data-type, and stored together with their header blocks. Other sets of experimental readings can be taken very easily as the sampling information is remembered and the Run number is automatically incremented, although the user has the option to change these parameters if desired. The table of header data is presented after each sampling run, as it is often required to change the positional information of some measuring instruments.

6.2 Ensemble average sampling

In wave and wave+current experiments (using regular waves) it is necessary to average sets of data from successive waves in order to extract information on the mean and fluctuating components at different phases of the wave cycle. In effect, data is required at definite points so that a mean and standard deviation can be calculated at each point. Due to the small variations in wave period it is impossible to synchronize sampling with the waves, even using the wave generating mechanism as a timebase. The ensemble average data acquisition routine uses a predefined water height as the start position of each wave cycle and realigns the

constant interval sampling to start at this point. This is shown in Figure 6. Using this method, variations in the wave period are automatically accounted for and phase errors virtually eliminated. The number of samples taken in each wave cycle are stored on a triggering data file, for later access by the ensemble averaging programs.

Before any data can be collected the trigger level has to be set; again, the user is aided by the computer program. Twenty waves are sampled to check that the water height gauge is adjusted, giving instructions to the user if it is not within certain bounds. The trigger level is set by an external control knob, but on the graphics terminal the trigger position is displayed as a moveable level in relation to the minimum, maximum and mean water heights. If the mean is selected as the trigger level then the user adjusts the control knob until the moveable line is level with the mean line. Having set the trigger level, twenty wave periods are accurately timed and this information presented to the user; who can repeat the setting up procedure or input the number of samples required per wave and the number of waves to sample. The ensemble average data acquisition routine then collects the data, which is stored as a temporary file on disc and processed in the same way as for steady flow sampling. Additional sets of experimental readings can then be taken very easily, with the option of resetting the trigger level or retiming the waves.

Two main data acquisition programs have been described briefly, showing how large amounts of data from many experimental runs can be collected and stored almost effortlessly, whilst the user retains the flexibility to get the experimental data required. Moreover, the data is stored in a systematic way so that, utilising the file management system, it may be efficiently retrieved for processing and graphical display.

7. DATA PROCESSING AND GRAPHICS

A large number of data processing and graphical display programs and subroutines have been written for this research project and therefore it is not intended to describe these in detail but to demonstrate how an effective file management system can greatly enhance the analysis and display of experimental data.

As shown in section 5.2 the programmer has routines available for generating standard filenames, updating the file directory and for checking if a file exists. These can easily be incorporated into any processing or graphics programs, giving semi-automatic file handling. The user input required will vary from program to program. For example, if a time series plot is wanted for any data then the user must specify the Test, Run and Datatype code. (The filetype must be 'D'). But if the Laser doppler anemometer data is being processed then only the Test and Run numbers need be specified - the program can automatically search for Datatypes L1 and L2. This facility is further extended by the use of 'multi-run' processing; the user specifies the Test (and perhaps the Datatype) and a range of Run numbers, in groups if necessary (eg 1-19, 23-46, 48-60). The program can then go ahead and retrieve the relevant data files, process the data and store the results. If a data file does not exist then a suitable message can be given and the processing can continue with the next set of data. This is particularly useful when large amounts of data are being processed after an experiment with many runs.

To illustrate the power of this system, let us look at the ensemble averaging of LDA and wave data. The program asks for the Test and also the Run numbers; from then on the processing and storage of results is automatic. For all the specified Run numbers the relevant data files are retrieved (Datatypes L1,L2 and W1) together with the triggering data (T___.W1). After extracting necessary information from the header blocks, the data is used to calculate seven sets of ensembled averaged data; namely horizontal and vertical velocities and turbulence intensities, shear stress, wave profile and its standard deviation. In addition, harmonic analysis of the ensembled data gives amplitude and phase information which is stored, with other details, in the header block. This is shown in Figure 3. Each of the seven sets of ensembled data + header block is stored as an 'E' Filetype with the appropriate Datatype code. This multi-run processing technique is used in any program which deals repetitively with sets of experimental data.

It is often necessary to summarise the results from a set of experiments and present this data in a tabular or graphical form. An example from this project is the production of mean velocity profiles

within a turbulent boundary layer. When the experimental data is initially processed, certain results are written into assigned locations in the header block attached to the data file. This is shown in Figure 3. The summarising program extracts these results, together with positional information, from specified experiments and creates a special 'summary file'. The summarised results can then be printed or plotted using a very flexible plotting program (detailed in the next section). Several types of summary files are produced depending on the type of experiment and the information required.

7.1 Graphics

A feature of the graphics programs is the flexibility of output together with the production of report ready graphs with excellent axes and anotation. The graphics routines are based on a modified subset of the Tektronix T.C.S. Using these graphic primitives, routines have been developed which can easily be incorporated into any program, making it straightforward to produce good quality graphs. Many programs give the option of drawing on the graphics screen or the plotter; thus allowing a relatively fast preview of a graph before getting a hard copy.

All the data required to create graphs is held in data files and accessed via the file management system. The header blocks attached to these files contain sufficient information for the automatic scaling of graphs if limits are not specified by the user. With time series plots it is possible to look at any particular part of the data series, with a maximum of 1500 data points across the screen or plotter. If more data points are required, the programs will divide the plotting area into two graphs. With the display of ensemble averaged data the graphics program has to be very flexible to cope with the many possible combinations of graphs. Up to four graphs per page (in three different sizes) can be produced with upto six lines on each graph, each one anotated. An example is shown in Fig.8.

The graphical output from summary files uses a particularly useful technique to specify the data to be plotted. Once the summary file has been retrieved, a menu of available information is displayed. An example of the steady flow menu is given in Figure 7. Each item or variable represents a series of results from the summarised experiments. A command string input is used to specify which

variables or combination of variables are to be plotted. Variables are enclosed in square brackets; constants may be input as integers or real numbers and arithmetic operators, including exponetial, may be specified. Some example command strings are given below.

```
[1]/[2]*0.138 , [12]/[6]

[5]**-0.5 , [4]+5.27

(x-axis)    (y-axis)
```

A special command string interpreter decodes the command and performs the arithmetic using the given variables and constants. Axis limits, labels, symbol type and curvefit options are available before the graph is plotted. An example graph is presented in Figure 9. Using this very flexible method of graph plotting almost all graphs can be drawn in final form without the user having to type in data. Therefore from experiment to final report all experimental data and results are stored in the computer, managed by the file management system.

8. ANIMATION OF EXPERIMENTAL RESULTS

During experimental work on waves, boundary layer velocity profiles are measured in great detail using the LDA equipment. Ensemble averaging the data produces graphs, as shown in Figure 8, describing the variation throughout a wave cycle of amongst other quantities, the horizontal velocity. This is at one point in the flow, and additional information can be derived from depth profiles plotted at different phases of the wave cycle. This is achieved by a program which retrieves ensembled data files at specified measuring positions, corrects for any phase differences and divides the wave cycle data into a given number of increments per cycle (typically 120). This processed data is then stored as a phase summary file and can be plotted in a similar way to other summary files. These phase plots give valuable information on the hydrodynamics of wave boundary layers but many graphs are required to get a full picture of what is happening during a wave cycle. A better approach has been to animate the phase plots so as to display a moving velocity or turbulence intensity profile. This is achieved using the facilities of the Manchester University Computer Graphics Unit, in particular their Vector General refresh graphics

screen and associate software support. Data in the form of phase summary files are transferred to the CGU via magnetic tape. The animation program displays the selected boundary layer profile as a graph changing with wave phase angle. A frame from the animation display is shown in Figure 10. The lower graph shows a velocity profile whilst the upper graph shows the phase point in relation to the wave surface profile. Among the options available interactively to the user are the control of animation speed including freeze frame, the production of graphic output files suitable for transferring the screen image to 16mm film, and the display of a theoretical velocity profile on the animation screen as a comparison to the experimental data. The theoretical profile, based on first order linear theory, is under real-time control. By means of control knobs the user can alter variables in the theoretical equations such as the β value, amplitude and phase with separate controls for the first and second harmonics. It has therefore been possible to dynamically compare experimental data with a theoretical analysis and demonstrate hydrodynamic behaviour previously unnoticed.

CONCLUSIONS

A data management system has been described which has proved most successful in dealing with large amounts of data from laboratory experiments. The acquisition and storage of data has been integrated with the subsequent processing and graphical display, utilising the file management system and associated routines for data retrieval and storage. This has enabled efficient data analysis and graphics programs to be written, with the emphasis on ease of use and flexibility of output.

REFERENCES

1. North West Consortium for Marine Technology - Project 2.2. University of Manchester. Funded by S.E.R.C.

2. Ellis, G.P., O'Connor, B.A., McDowell, D.M. (1981). Generation of laboratory waves using a microcomputer. 2nd International Conference on Engineering Software, London.

FILETYPE "D"		FILETYPE "E"
INTEGER HEADER	REAL HEADER	REAL HEADER
1- Records	Calibration	1- Wave period
2- Record length	Sampling rate	2- Wave amplitude
3- Date	Position:x	3- Wave phase
4- Test	y	4- Wave length
5- Run	z	5- Position:x
6- Samples	Mean 1	6- y
7- *	Mean 2	7- z
8- *	Turbulence 1	8- Mean
9- *	Turbulence 2	9- Min
10- *	Shear	10- Max
11- Datatype	Min 1	11- Amp/wave amp
12- -	Max 1	12- Phase-wave phase
13- -	Min 2	13- -
14- -	Max 2	14- -
15- Zero	Depth	15- Water depth
16- -	Roughness	16- Roughness
24- Sequence no.		17- -
25- Exist flag		18- Date

Figure 3. Example of header block formats

FILENAME	DATE OF TEST	TYPE OF FILE	TEST NUMBER	RUN NUMBER	DATA TYPE-CHANNEL	NO.OF RECORDS	RECORD LENGTH	ON TAPE
RK2:D06002.L1	17-09-82	TEST DATA	60	2	LASER #1	23	333	NO
RK2:D06002.L2	17-09-82	TEST DATA	60	2	LASER #2	23	333	NO
RK2:D06002.W1	17-09-82	TEST DATA	60	2	WAVE #1	23	333	NO
RK2:E06002.S01	17-09-82	ENSEMBLED	60	2	SHEAR #0	1	238	NO
RK2:E06002.T11	17-09-82	ENSEMBLED	60	2	TURBLNCE#1	1	238	NO
RK2:E06002.T21	17-09-82	ENSEMBLED	60	2	TURBLNCE#2	1	238	NO
RK2:E06002.V11	17-09-82	ENSEMBLED	60	2	VELOCITY#1	1	238	NO
RK2:E06002.V21	17-09-82	ENSEMBLED	60	2	VELOCITY#2	1	238	NO
RK2:E06002.W11	17-09-82	ENSEMBLED	60	2	WAVE #1	1	238	NO
RK2:E06002.X11	17-09-82	ENSEMBLED	60	2	WAVE SD #1	1	238	NO
RK2:R06002.SM1	22-09-82	SUMMARY	60	2	NUMBER #1	22	52	NO
RK2:T06002.W1	17-09-82	TRIGGERING	60	2	WAVE #1	2	70	NO
RK2:W06002.PH1	23-09-82	PHASE SUMARY	60	2	NUMBER #1	12	80	NO
RK2:W06002.SM3	23-09-82	WAVE SUMMARY	60	2	NUMBER #3	33	62	NO

14 FILES

Figure 4. Listing from the file management directory

CHAN. NO.	DATA CODE	POSITION (LENGTH	WIDTH	HEIGHT)	CALIB.	ZERO
0	W1	0.00	250.00	19.00	0.00470	1904
1	L1	0.00	250.00	7.19	0.32935	1413
2	L2	0.00	250.00	7.19	-0.35075	1380
3	S1	50.00	300.00	10.00	0.00173	878

(maximum of 16 input channels)

Figure 5. Interactive input of header information

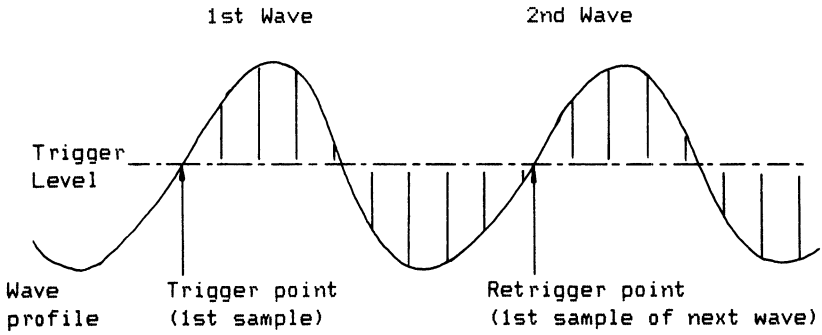


Figure 6. Ensemble average data acquisition

```
DATA ELEMENTS AVAILABLE:
 1- VMEAN          2- HMEAN
 3- VTI            4- HTI
 5- SHEAR
 6- DEPTH          7- ROUGHNESS
 8- SHEAR VEL.     9- VEL.GRADIENT.
10- POSITION: ALONG      11- ACROSS      12- HEIGHT
13- RUN NO.       14- TIME

#### COMMAND STRINGS ####

DATA ELEMENTS ENCLOSED IN [ ]   EG. [12]
ALLOWABLE OPERATORS: ** * / + -
DELIMITER BETWEEN STRINGS:  ,
ANY CONSTANTS ALLOWED
NEGATIVE NUMBERS ALLOWED
EXPRESSIONS EVALUATED FROM LEFT TO RIGHT
```

Figure 7. Menu from summary file plotting program and command string input instructions

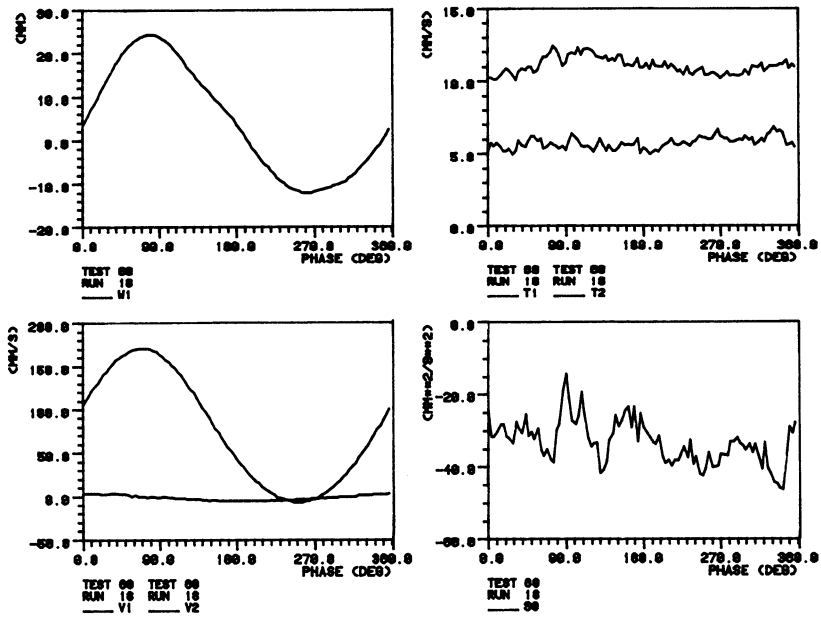


Figure 8. Graphical output of ensemble averaged data

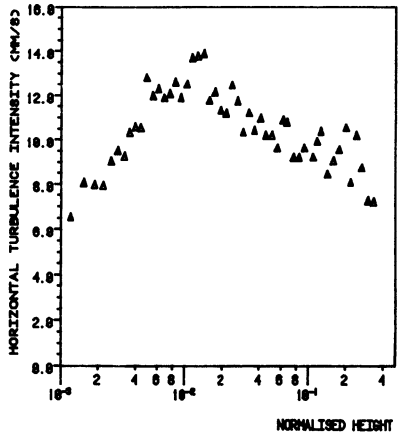


Figure 9. Example of summary data graph, using command string to specify axes. (Graphs normally produced in A4 size)

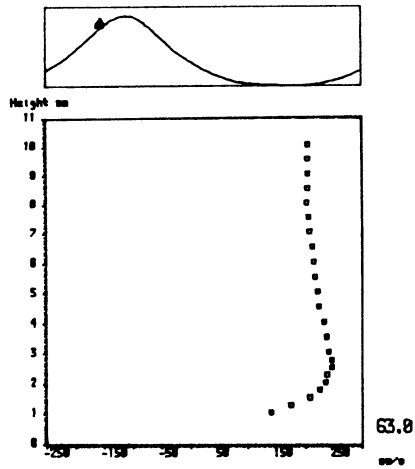


Figure 10. Frame from a 16mm animation film showing the boundary layer velocity profile at the 63 phase point

AN ENGINEERING DATA MANAGEMENT SYSTEM FOR COMPUTER AIDED DESIGN

F.J. Heerema, National Aerospace Laboratory NLR, The Netherlands

H. van Hedel, Control Data B.V., The Netherlands

ABSTRACT

During design processes experiments and computational tools are increasingly important. As a result large volumes of data come available. Engineers need a means to cope with:
- managing the data flow between the computational aids and experiments;
- extracting the, for interpretation relevant, information from the data coming from the various sources.

Computer aided design techniques can be used to satisfy this need. The paper deals with the Engineering Data Interactive Presentation and Analysis System (EDIPAS), which is a tool for both data management and data analysis and presentation.

1. INTRODUCTION

During design processes, experiments and computational tools are alternately applied. For example, to decrease fuel consumption of aircrafts the design of the shape is optimized with respect to the drag. Complex computational methods are used to calculate the aerodynamical quantities for a given shape and to determine the shape for prescribed aerodynamical properties. Wind tunnel experiments are used for verification in the design process and for detailed investigations to support the development of the computational methods.
Enhanced measuring techniques and continually improving computational methods, made possible by the still increasing power of computers, produce an ever increasing amount of data.
In this kind of situations an infrastructure is necessary (Loeve, 1982), to provide engineers with the needed information, which meets two major requirements:
- the management of the data exchange between the various experimental and computational processes;
- to trace and to use the information relevant for interpretation.

Computer aided design techniques, like database management and

interactive graphics, are able to construct solutions which meet these requirements. To construct a solution for each occasion gives rise to the following disadvantages:
- engineers have to spend considerable amounts of time to design the database to be used;
- in each situation special purpose software has to be developed to interface experimental facilities and the various computational processes with the database;
- basic data analysis and graphical presentation software will not be generally applicable.

To overcome these disadvantages an engineering data management system has been designed and made operational at the National Aerospace Laboratory NLR (The Netherlands). This system is known under the name EDIPAS ("Engineering Data Interactive Presentation and Analysis System"). EDIPAS serves a wide range of engineering and computer aided design applications.
In order to reach a broad user community the package is marketed by Control Data.
This paper deals with the concepts applied in the design of this system. The available facilities and some examples of applications with the system are presented.

2. SYSTEM OVERVIEW

In the Introduction, EDIPAS is presented as a system that enables engineering data management, and engineering data analysis and presentation. Before the technical concepts are described in more detail, an overview of the system characteristics is given.

2.1 General concept

To meet the requirement of tracing and using the data coming from the various computational and experimental sources, these data have to be treated as one single source. For that reason the facilities of EDIPAS are built around a database, in which the information of computations, geometry-handling tools, and experiments can be stored (Fig.1). To control this database, a commercial database management system is used ("Information Management Facilities", IMF, of Control Data). To save development time, this database management system has been selected in the second half of the seventies, to support several technical applications at NLR. In order not to bother engineers with problems of database design, the data management part of EDIPAS enables engineers to apply naming conventions they are used to, for data items and datastructures. A database may be started for just one project or may contain data from several projects. The database manager of a project decides who has access to the database under his control.
The information in the database is used to exchange data between the various processes. The data analysis and presentation part of EDIPAS is used to receive the required information from the database and present this information graphically or as printed reports, in completely user-defined layouts.

2.2 Main system components

EDIPAS consists of a number of subsystems, each of which comprises functions. A function may be a computer program or a part of that. Figure 2 gives an overview of the system.
The main system components to be mentioned here are:

a. the data management part:
   - a utility to intialize a project database;
   - input programs to load data from files into the database under user control;
   - utilities to dump parts of the database to off-line devices (e.g. magnetic tapes) and to re-load data stored in this way;
   - a set of routines to extract data from and to load data into the database in a controlled way;

b. the data analysis and presentation part:
   - a command-driven interactive subsystem with functions to select information, to build curves and arrays of curves, to smooth and to interpolate (by means of spline algorithms), to define and to display graphs with user-controlled layout and annotation, and to define tables and layout for printed reports. Upon user request, commands may be grouped and edited to parametric procedures, which can be stored in the database to be used in later interactive sessions or executed in batch.
   - a program to prepare the control sequence for the execution of (pre-stored) procedures in a batch-run of the interactive sub-system.

2.3 Operational environment

As a consequence of the state of the art of database management and data-communication between computers, NLR has decided to install EDIPAS on the central computing facility (a Control Data Cyber 170-855) of the computer and terminal network (Loeve, 1976) shown in figure 3. An example of routine operation with EDIPAS can be given for data from measurements in the High-Speed Wind Tunnel (HST). The measured data is pre-processed locally on a mini-computer to obtain aerodynamic coefficients, which are sent as file from the location in Amsterdam to the central computer, located in the Noordoostpolder. The data are stored in an already initialized EDIPAS database. In the same job-run, pre-stored procedures can be executed to obtain graphs with predefined layouts, which are chosen as standard for the type of measurement.
The file containing the graphs can be handled in two ways. First, it can be inspected with a Tektronix graphic terminal configuration, connected to one of the communication front-ends of the network (Fig.4). Secondly, the file can be sent to one of the plotters available on both NLR locations, to obtain reproduction-quality graphs. It still remains possible for engineers to inspect the created data base interactivly.

2.4 Examples of application

Figures 5 and 6 give an impression of the variety of applications of EDIPAS.
Figure 5 presents the geometry of a blade of a ship propeller. The data describing the various sections and the leading and trailing edges of the propeller, are represented by splines.
Another example, taken from computed and measured data, is given in figure 6. The solid line represents the experimental results, the various curves, represented by means of symbols, show computational results. This example is in the area of boundary layer calculations. Results of calculation methods applied at several international institutes are compared with those of a reference experiment (v.d. Berg, et al, 1983).

3 INFORMATION CONCEPT

3.1 Database organisation

In order to make EDIPAS widely applicable, some decisions had to be taken during its design. These design decisions were related to project interference, data structures, data entities, and naming conventions for the data items.
First, it was decided to create the possibility of assigning one database to each project. The definition of a project is left to the user community. In order not to burden users with technical questions related to database management and with a cumbersome information analysis for each project, an database organisation has been set up using the NIAM-technique. This organisation, kept in the meta-database, is the same for all EDIPAS project databases. This is made possible by means of Control Data's database management system IMF ("Information Management Facilities"). IMF enables the control of several databases with the same meta-data. Data items are stored alphanumerically (i.e. floating-point numbers and strings), in order to realize the required accuracy for the technical environment in which EDIPAS is used.
Further, the naming of data structures and data entities is completely left to the users. Engineers can access the database with naming conventions they are used to. To simplify the data retrieval process, EDIPAS allows users to employ one or more hierarchical identification levels, with user-defined level names. At the lowest level, the elementary engineering data is stored in entities called datablocks. In figure 7 the hierarchical identification levels are presented, together with some examples.
The elementary engineering data, measured or calculated quantities, are uniquely identified for retrieval by means of the hierarchical level identification and the user-provided name of the quantity (Fig.8).

3.2 Basic data entities

In an EDIPAS database the following basic data entities are supported:

a. Datablocks. A datablock is an assembly of coherent engineering data, which means that the data items in a datablock belong together in whatever way a user wishes (e.g. results computed with one sample input). A datablock allows matrices, single values and characteristic values as data elements:
   - Matrices. A matrix is a collection of values with the same dimension and the same meaning (e.g. measurements of a quantity at successive time steps). One- or more-dimensional matrices are allowed. The dimension of the matrix is defined by the user, according to his view on the data.
   - Single values. A single value is a quantity which is allowed to occur once within a datablock. As an example: single values can be used to describe the conditions (temperature, speed) under which measurements in the matrices of the same datablock,are taken.
   - Characteristic values. A characteristic value is a descriptive value for the entire datablock. Characteristic values are user-defined and may be measured or calculated. The set of characteristic values of a datablock describes that block uniquely.

   The values of the data elements may be either floating-point numbers or strings. The various values of a matrix all have to be of the same type. Each occurrence of a data element is identified by a user-defined name (Fig.9).

b. Structures. Structures are a means with which the user can describe his view on the data in terms of relations between an hierarchy of his datablocks. The user defines this view with various levels. The datablock characteristic values may correspond to these levels. In that case, the datablock characteristic values describe the place of a datablock in the user's view. More then one structure is allowed in a database. A datablock may belong to one or more structures, but it has to be unique within each structure.

3.3 Interpretative entities

From the basic entities mentioned above, a user can construct curves and arrays of curves as interpretative entities:

a. Curves. A curve is constructed from a user-selected ordered set of pairs of values (points or independent and dependent variable). A curve can be either the original set of points or an analytical form, based on a B-spline algorithm (Dierckx, 1975; Renes and Beintema, 1980). A curve is uniquely defined by a set of specific values. These values consist of the structure level common to the originating data, a set of either automatically determined or user-defined curve values and the user defined curve name.

b. Arrays. An array is constructed from a user selected ordered set of curves, or the result of a cross-interpolation of an existing array. The array is, like a curve, uniquely defined by the common level of the curves that constitute the array,a

set of automatically determined or user-defined array values, and the user-defined array name.

3.4 Supporting entities

The presentation and interpretation functions of EDIPAS are supported by the Picture and Procedure entities:

a. Picture. A picture is a set of instructions that describes a graphic display screen image as defined by the user. This picture contains all the information the user has specified to obtain such a screen image. The entity is used to store a picture as a result of a screen-editing operation, to be able to reproduce or to extend the picture at a later time.

b. Procedure. A procedure is an ordered set of EDIPAS-commands, specified by the user, which can be executed under parameter control. Procedures can be used in three different ways:
   1. as a short description of a fixed set of consecutive commands for ease of use;
   2. as a description of a complete data manupulation process under user control;
   3. as a means to describe a batch process.

   Procedures can be created either by means of a procedure-editor or as a results of a part of an interactive session. The latter method assures that the proper syntax and command sequence is used.

For operational reasons the database is used to store some additional information sets. These information sets are used to load and retrieve data, to keep track of default setting of command-parameters during interactive sessions, and to maintain commands and messages. In the next sections the relevant information sets will be discussed in their operational context.

4 SOFTWARE CONCEPTS AND COMPONENTS

4.1 Development approach

From the beginning of the development of EDIPAS it was clear that it had to be an open-ended system. The evolution of the system would proceed, once an operational version would have been taken into production. To make this evolutionary development possible and to enable easy maintenance, a set of general applicable subroutines has been developed to support the database modification functions. This set of subroutines are the interface between the database management system IMF and the EDIPAS functional modules, that result in a modification of the database contents. The routines, like a wrap round the database, serve also as a means for comunication between the various functional modules of EDIPAS (Fig.10). EDIPAS is composed of three subsystems, each comprising a number of functions.

4.2 Database administrator support

To support the database administration activities five facilities are to be mentioned:

a. Initialization of the database. Descriptive informations is stored about the project, the person responsible for the database and users who are granted acces to the database.
b. Definition of datastructures. Data structures to be used for loading and retrieving information can be defined in advance.
c. Deleting data structures. Previously loaded structure definitions, not to be used anymore, can be removed from the database. The datablocks of that structures still remain active in the database.
d. Audit of the database contents. An audit facility can be used to present an overview of the complete contents of a database or a user defined part of it.
e. Back-up facility. The back-up facility provides for the selective dump of data-base contents to an off-line storage device, such as magnetic tape. Re-load of dumped information or a selection of it, is possible either into the original database or into the database of another project.

4.3 Data management

There are three ways in which the database can be accessed for storage or retrieval of data, except for the analysis and presentation part of EDIPAS.

a. Loading of data in the database. In order to load data into the database from a sequential file, the description of the contents and the format of this file has to be known to the system. This file description, called a macro, in loaded into the database to be used each time a file is to be loaded according to the file description. At load-time directives are used to combine the file description and the data file, and to specify the structure to be used.
b. Retrieval of data from the database. In principle this is the revers of the load process. Again the description of the output data file has to be known to the system and is stored in the database.
c. General application interface. A set of interface modules is available to support direct communication between the database and special purpose application programs. The Fortran-callable interface modules guarantee safe load and retrieve actions in user programs.

For the maintenance of the database contents, the data management subsystem contains a function to inspect, change, delete, and add data items. Simple arithmetic operations with matrices are allowed.

4.4 Analysis and presentation

For interpretation purposes, a number of analysis and presentation functions are available (Fig.11). Each function operates on the contents of the database and produces a result that may be stored in the database upon user request. The functions are

part of an EDIPAS subsystem, which can be executed interactively or in batch. The sub-system is command driven. The analysis and presentation functions are:

a. Consult. With the Consult function the user is informed about the stored data entities. Modification of the data base contents is not supported.

b. Table. A table is a matrix of values, each column representing a set of data of the same quantity. The function Table allows to specify and to select data from the database.

c. Curve. With the Curve function the user selects an ordered set of pairs of values to construct a curve. A variety of selection criteria is available. Arithmetic operations with curves are supported, e.g. calculating the difference of two curves.

d. Smooth. With the Smooth function the user represents a (previously selected and stored) curve analytically by means of B-splines (Dierckx, 1975 ; Renes and Beintema, 1980). The user has a number of facilities to control the curve fitting process. The obtained spline approximation can be added to the already stored curve data.

e. Array. With the function Array the user combines selected curves to a set, called an array of curves. Selected combinations can be modified to new arrays.

f. Cross. Within the Cross function the user obtains a cross interpolated array of curves from a selected array. The process is best illustrated by figure 12. The array of C_L-alpha curves on the left has the speed (Ma) as curve value. The result of the cross-interpolation is shown in the right-hand picture, where the C_L-Mach curves are given with the angle of attack, alpha, as curve value. This example is taken from a EDIPAS application for windtunnel measurements.

g. Picture. The Picture function allows the user to describe a complete screen image element by element,either to be displayed on a graphical terminal or on an offline plotter. A picture consists of one or more windows. Each window may contain x-axis and y-axis (as many as required), curves, arrays, and the associated values. Beyond the area of a window, legends and annotation can be added. Figures 5, 6 and 12 present examples of the possibilities of this function.

h. Report.With the Report function the user creates a tabular printout of previously defined and stored tables. The user specifies the layout of the printed report by means of commands, defining headers, footers, etc.

i. Procedure. With the Procedure function the user combines commands available for the subsystem to parametric procedures. Procedures can be stored in the database for later use, which may be either interactive or batch execution. Existing procedures may be altered by means of an edit facility.

During interactive execution of the sub-system, it is possible to log the commands used. The result is a procedure that has been proven to be correct. This procedure can be used in the same way

as those built in the Procedure function. For batch processing an interactive program is available to create an input file that will execute the required procedures.

5 OPERATIONAL ASPECTS

5.1 Datamanagement

Operations with the data management part of EDIPAS is best illustrated with an example. Figure 13 presents the actions necessary to initialize an EDIPAS database and to load data originated by an auxiliary user program.
The organisation of the database (the meta-data) is described once for all in terms of the IMF Data Definition Language. At initialization time, a project database is created, containing this meta-data. The initialization program (INIT-DB) adds the user specified information about persons having access to the database, together with already known names of the structures and their levels.
In order to make data loading to a routine operation, the user has to analyse his data with respect to the hierarchical information structure, and to the naming conventions to be applied for the structures and the engineering data elements. As a result of this, the description of the data file is composed, which is called an EDIPAS load-macro. The syntax of the load-macro is checked and, when no errors are found, the macro is stored in the database. The user program has to facilitate the generation of the actual data file, according to the description of the load-macro. To load the data file, a series of load instructions is used, which contain the name of the macro to be used and specify characteric values with which the data are placed in the data structures. For routine operations the user program has to generate the load instructions as well.
When more user programs originate data for the EDIPAS database each different data file needs its own description, recorded in load-macros.
For data extraction from the database, to obtain interface datafiles to user programs, an analogous (but reversed) process has to be used. Again using a macro, describing the resulting data file and instructions to extract the data.

5.2 Analysis and presentation

The analysis and presentation part of EDIPAS is a command-driven subsystem. To facilitate the interactive usage of this subsystem, the parameters of the commands for spline approximation and for presentation have a system default setting. The user can replace the system default setting subsequently by global values for all functions and by local values for a specific function of the subsystem. Having specified a local default setting, the user may overrule this temporarily during the execution of one command by specifying the appropriate command- parameter values. The global and local settings can be stored in the database, as session information to be used when re-activating the originating session.

During an interactive session commands can be grouped into parametric procedures, which can be used for later sessions or for batch execution. The parameters of these procedures have as default values those used during the generating interactive session. These parameter default values may be replaced by actual values when a procedure is re-activated.
Pictures and reports, being sets of commands describing graphical and printed layout for presentation purposed, are stored in the database with user-defined names. These may be used repeatedly either in interactive sessions (graphical presentation) for direct interpretation or in batch. The batch runs are mainly used for routine production of larger amounts of graphs and the printed reports. Figures 14 and 15 give examples of thus produced graphs. Figure 14 is an example of time-series measured during landing of aircraft. Figure 15 is an example of pressure measurements performed on an aircraft wing in the High- Speed Wind Tunnel of NLR.

6 FUTURE DEVELOPMENTS

As mentioned before, EDIPAS has been designed as an open-ended system. Developments will proceed continuously in the near future. The developments will meet the still evolving user requirements for additional facilities and improvements of existing functions. Except for improvements, attention will be paid to the following features:

a. extension of a structure re-definition facility, providing an enhancement for users, to manipulate with the data structures and their levels;
b. extension of the authorization from database level to the level of data elements, which gives a user explicit access to an operation with a specific data element;
c. logging of transaction on the database, to keep track of all or part of the activities with respect to a project database;
d. re-establishing of a database to a previous condition by means of the logged transaction;
e. archive facility to keep track of contents of magnetic tapes with selectively dumped engineering data and to serve quick re-load of these data.

7 CONCLUSIONS

EDIPAS meets the requirements of managing the data flow between various computational and experimental processes and of analysing the data coming from the processes. The neutral database concepts enable a wide range of engineering applications. Powerful facilities such as procedures of commands and user-controlled definition of graphs and printed reports, serve routine operation.
The general concepts applied in the design of this system made it to an indispensable element in the infrastructure of NLR for computer aided design.

8 ACKNOWLEDGEMENT

The authors gratefully acknowledge mr. H.A. Kreykamp of NLR, for his contribution to the completion of this paper and his critical reading of the text.

9 REFERENCES

Berg, B. v.d., Humphreys, D.A., Krause, E. Lindhout, J.P.F. (1983) Comparison of Three-dimensional Turbulent Boundary Layer Calculations for Five Experiments. The Eurovisc 1982 Berlin Workshop. Friedr. Vieweg and Sohn, Braunschweig/Wiesbaden (to be published).

Dierckx, P. (1975). An algorithm for smoothing, differentiation and integration of experimental data using spline functions. Journal Comp. Appl. Math. vol.1, 3: 165-184.

Loeve, W. (1976). A Hierarchival Network Linking Two Research Laboratorium. Computer Networks (1976), 1 : 119-129.

Loeve, W. (1982). An Infra-structure for Computational Fluid Dynamics to serve Computer Aided Design. Symposium for users of finite element methods, Delft Technical University, NLR MP 82046 (Paper in Dutch).

Renes, J.J. and Beintema, D.A. (1980). The use of B-spline Approximations and of an Array Processor in the IRAS Ground Operations and Preliminary Analysis Facility. 3-RD BIS Conference on Computers in Space Applications, NLR MP 80009 U.

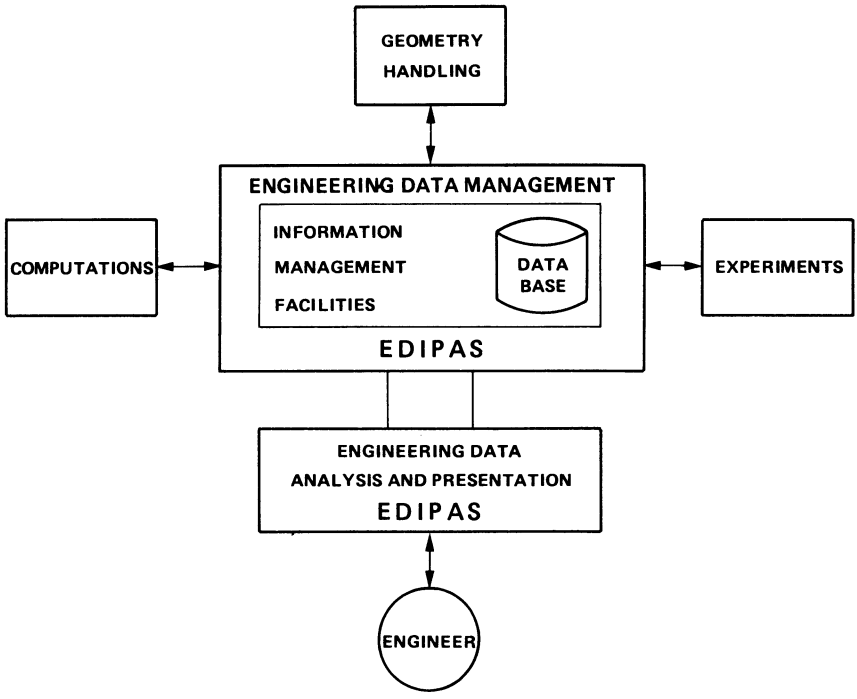


Figure 1. Architecture of an infrastructure using EDIPAS

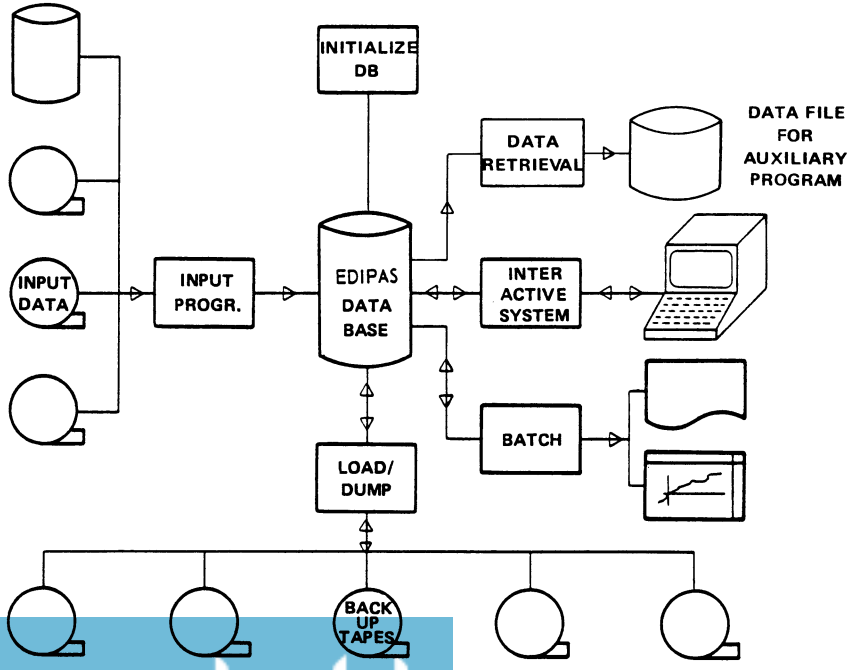


Figure 2. EDIPAS—system overview

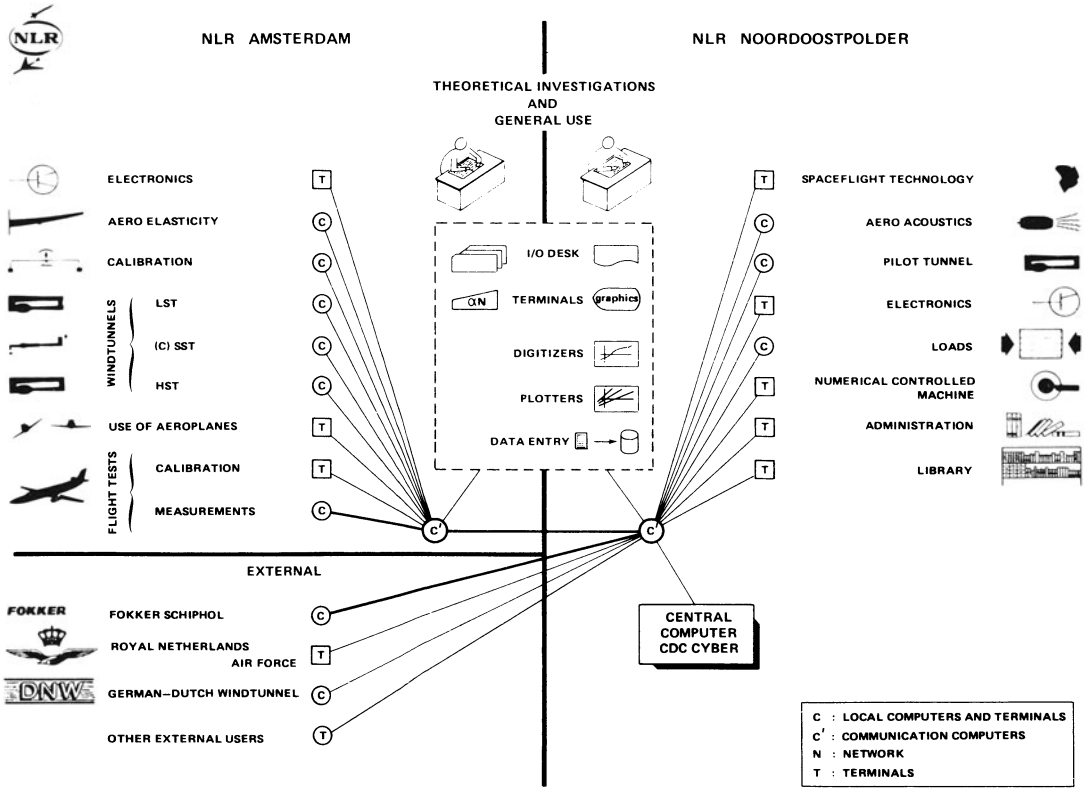


Figure 3. NLR computer and terminal network

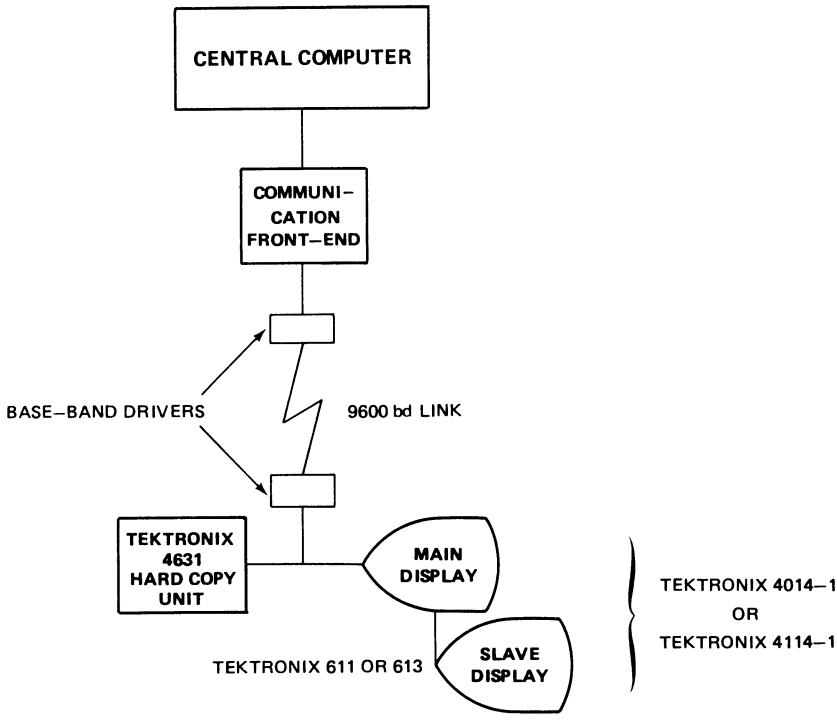


Figure 4. Graphic terminal configuration

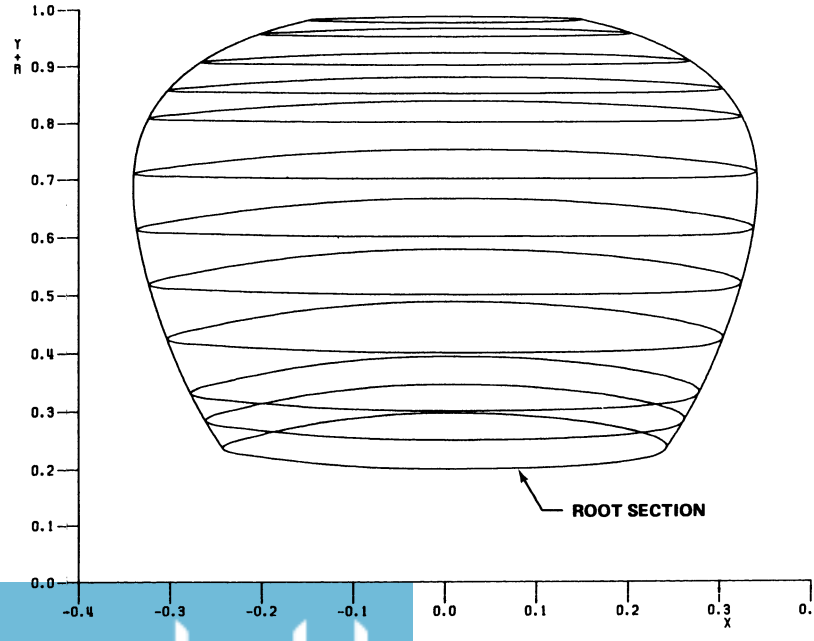


Figure 5. Geometry of ship propeller blade

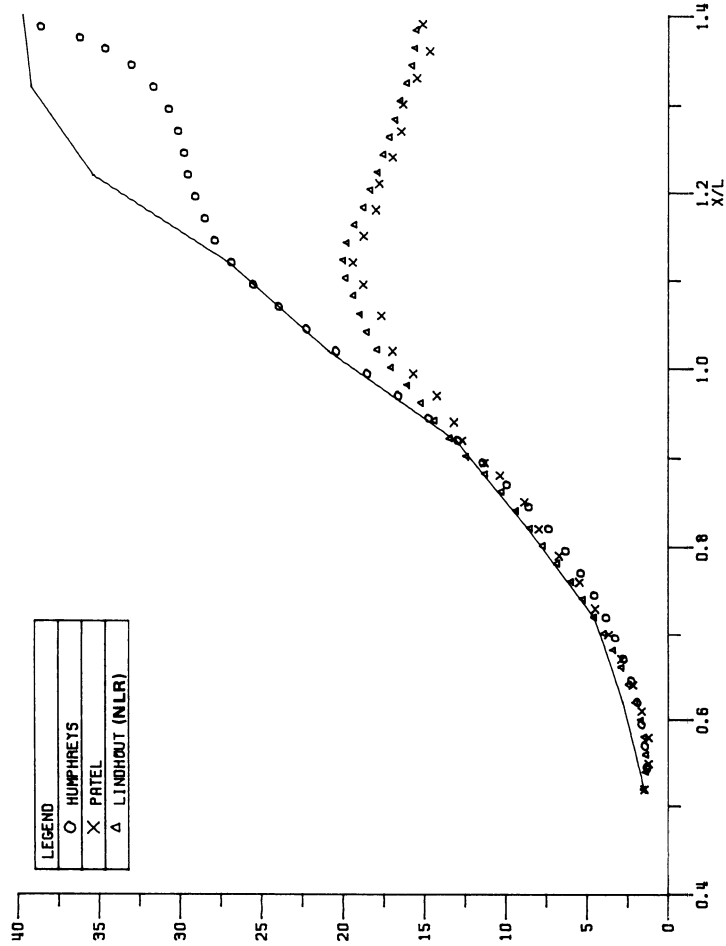


Figure 6. Comparison of computational and experimental results

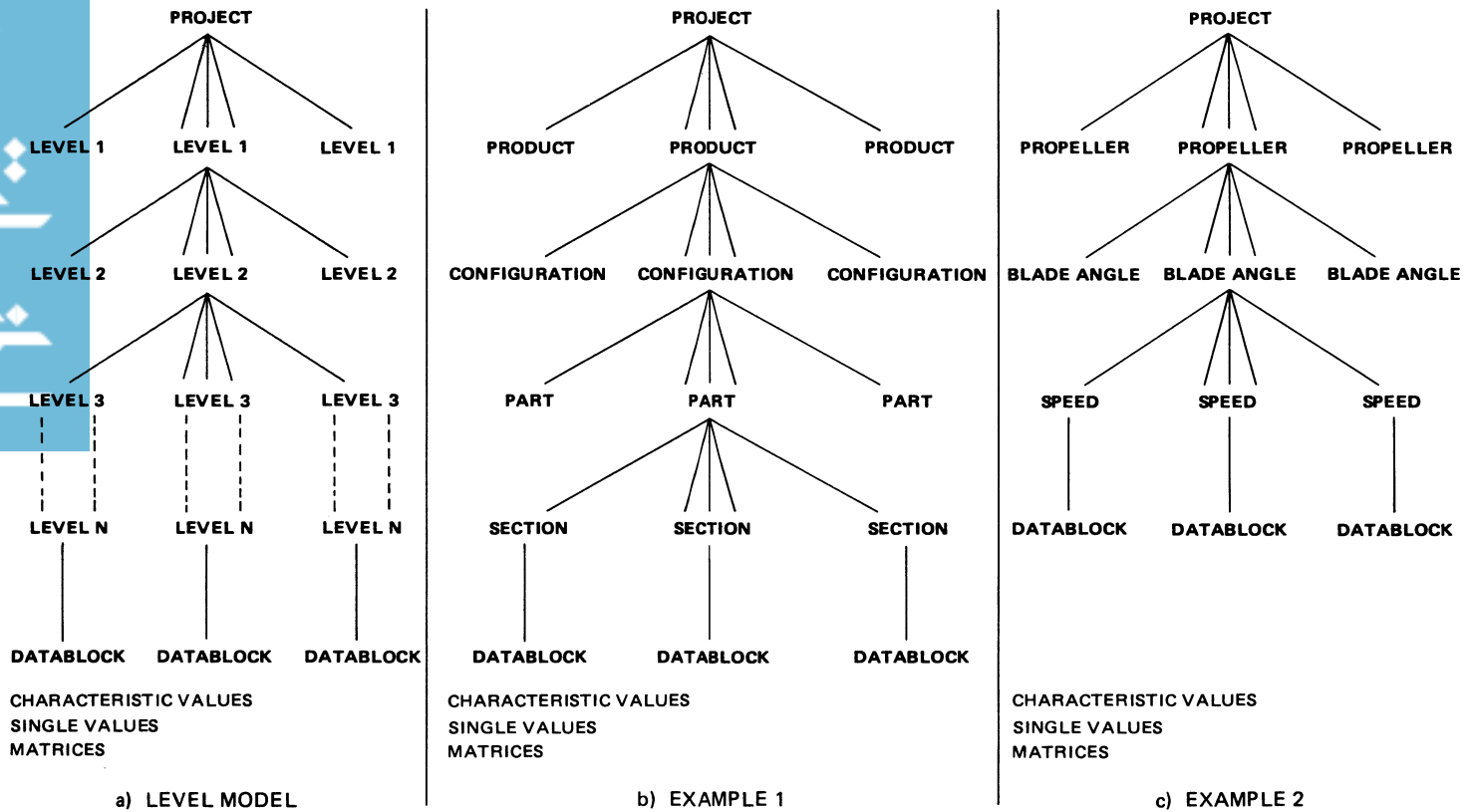


Figure 7. Hierarchical identification levels

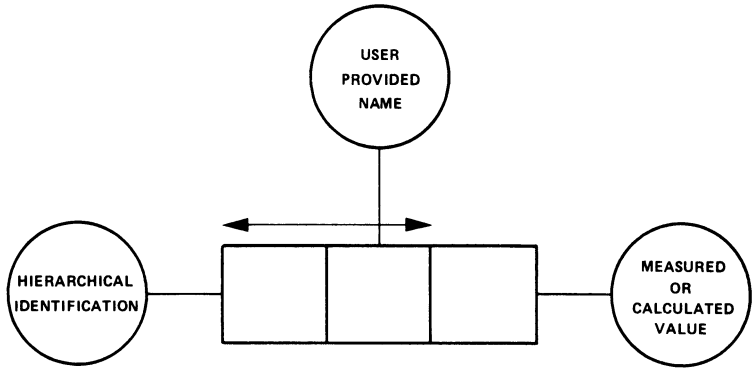


Figure 8. Basic information retrieval concept

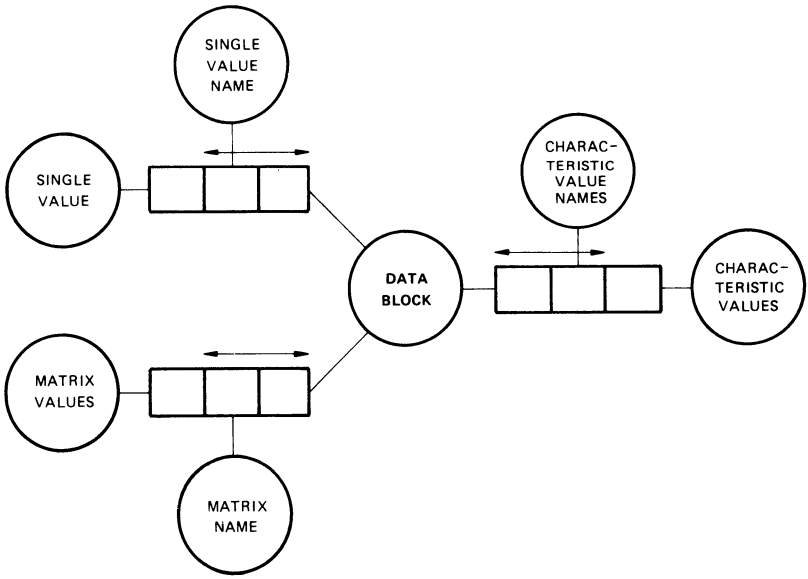


Figure 9. Elements in a datablock

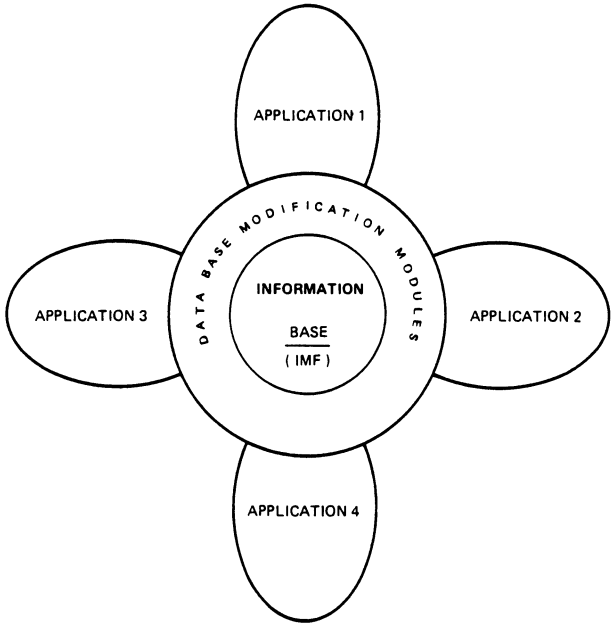


Figure 10. Modular approach of the system

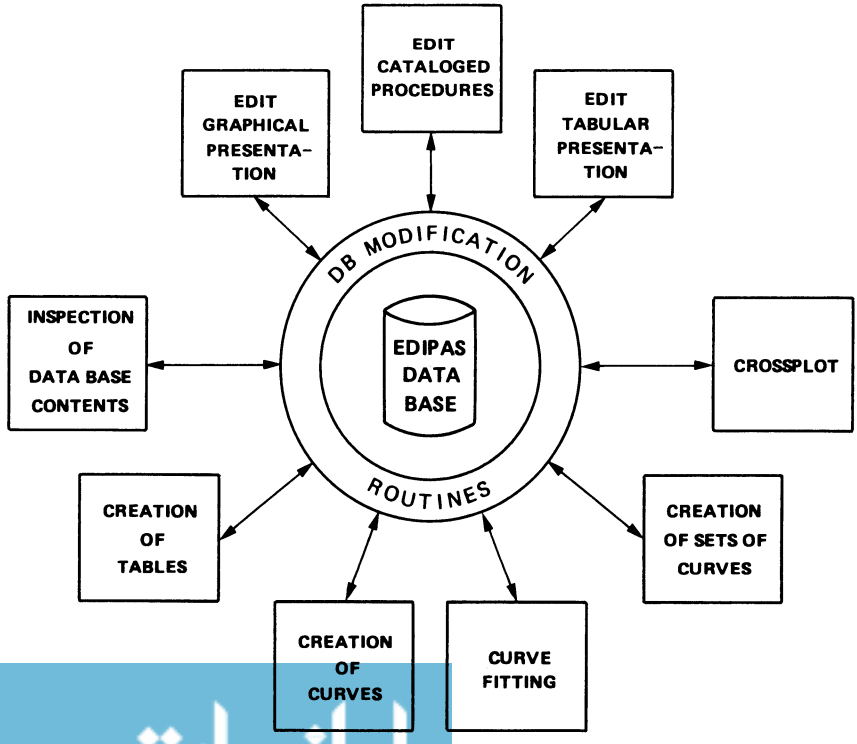


Figure 11. Overview of the interactive subsystem

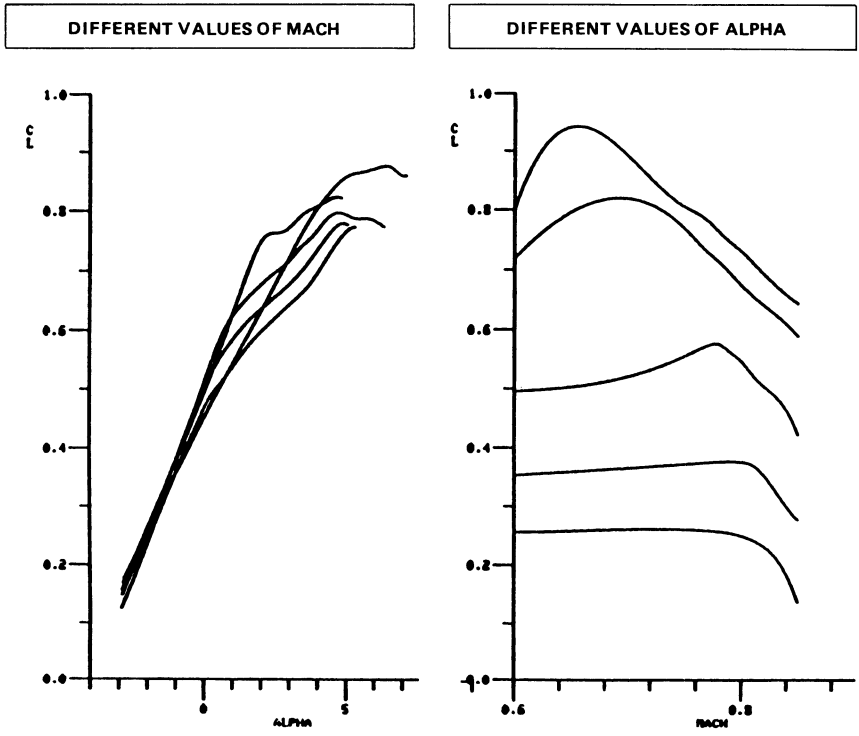


Figure 12. Example of the Cross interpolation function

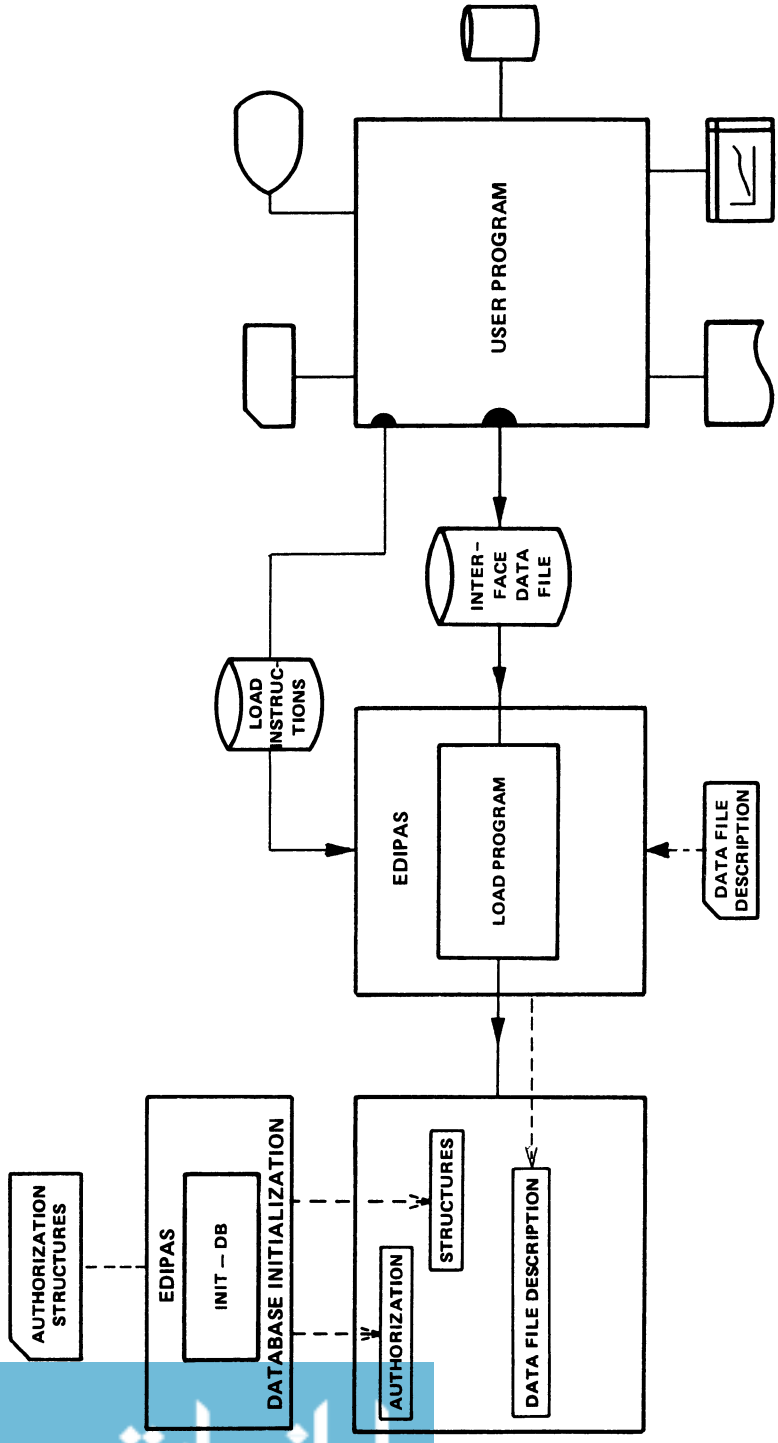


Figure 13. Activities to load from an auxiliary user program

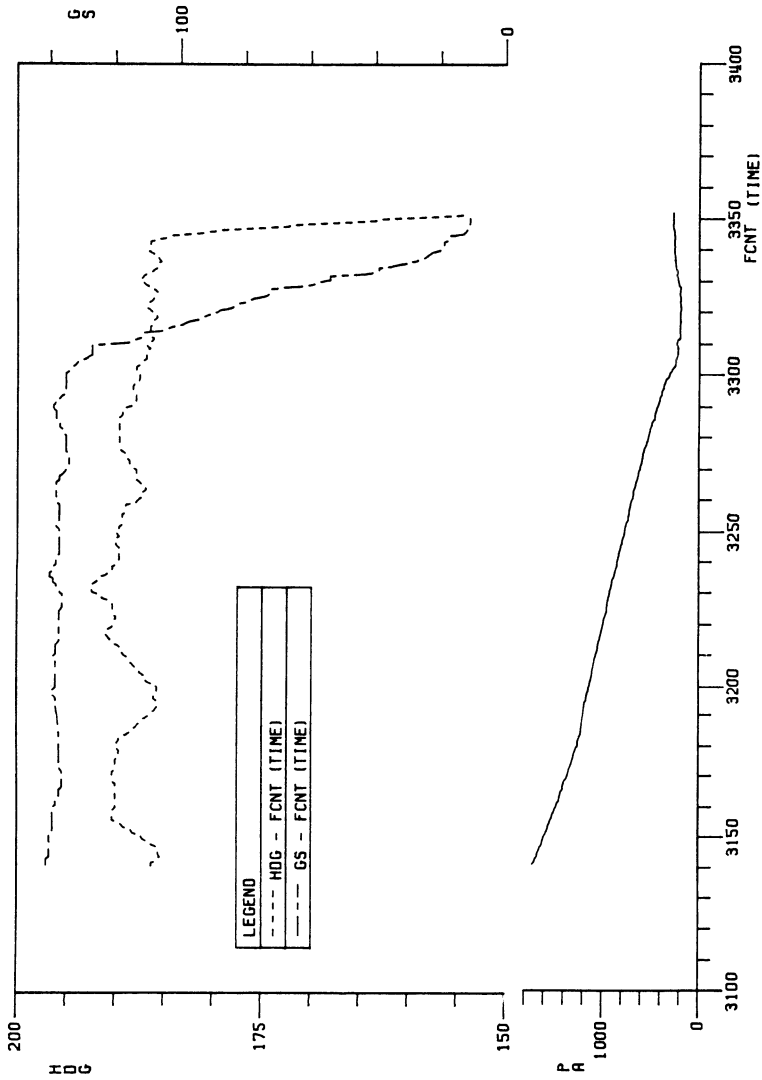
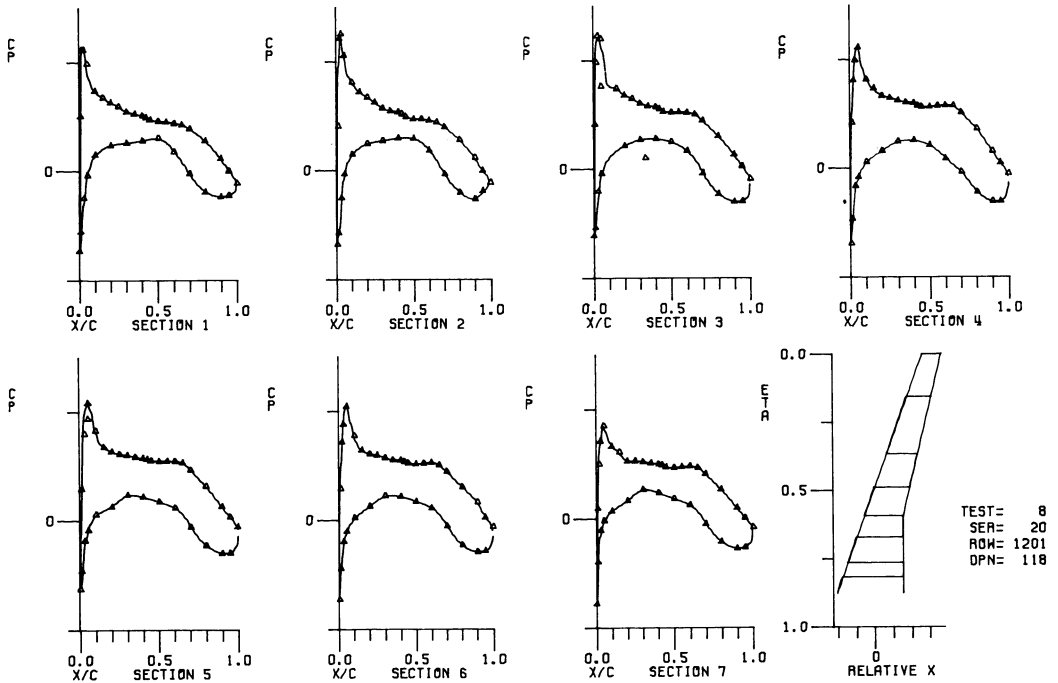


Figure 14. Example of routine produced graph of time–series



PRESSURE DISTRIBUTIONS OF WING MEASURED IN NLR HST 1980, MACH = 0.60, ALPHA = 0.463, Re = 2980000

Figure 15. Example of routine produced graph of pressure measurements

FINITE ELEMENTS

A PLATE BENDING ELASTO-PLASTIC EQUIVALENT GRID FINITE ELEMENT

William McCarthy
Leonard Traina
Abdul Melhem

New Mexico State University

INTRODUCTION

A bridge analysis computer program BRANDE IV (McCarthy, White, Minor 1980) was developed as an instrument for the limit analysis of a bridge superstructure. BRANDE IV provides an elasto-plastic analysis which incorporates a convenient data format and reduced input work requirements. The program is readily adaptable to both continuous slabs and slab over girder-diaphragm networks. The program incorporates an elasto-plastic finite element analysis of plates as the primary component of the limit analysis procedure.

An elasto-plastic analysis of plates is a complex affair using standard finite elements. A grid-framework model that simulates plate behavior, first proposed by Yettram and Husain (June 1968) and later by Traina (1968), makes possible a less complex limit analysis approach developed through the Engineering Foundation (McCarthy 1982). A limit analysis elasto-plastically analyzes a structure through a series of elastic cycles. Plastic hinges are inserted into the structure at the end of each cycle at points where the bending moment reaches the plastic moment capacity. The grid-framework model is easily adaptable to a limit analysis procedure because the model consists of plane-grid members which have an easily definable elasto-plastic history.

The grid-framework plate element is readily superpositioned on plane-grid members for the analysis of structures consisting of a slab over girder-diaphragm network. Another advantage is that it provides a more concise definition of plastic flow due to the large number of potential plastic hinge points associated with the grid-framework element. This is particularly important for the many instances where coarse grids are sufficient to generate reasonable results.

The Grid Framework Element

The grid-framework element illustrated in Figure 1 consists of four plane grid memebers framing two diagonal plane grid members. Grid members are assigned moment of inertias, I_a, I_b and I_c and torsional constants, J_a, J_b and J_c, of magnitudes that force the element to behave as a plate. The correct magnitudes are determined by compatibility at the corners between a typical plate and grid-framework element.

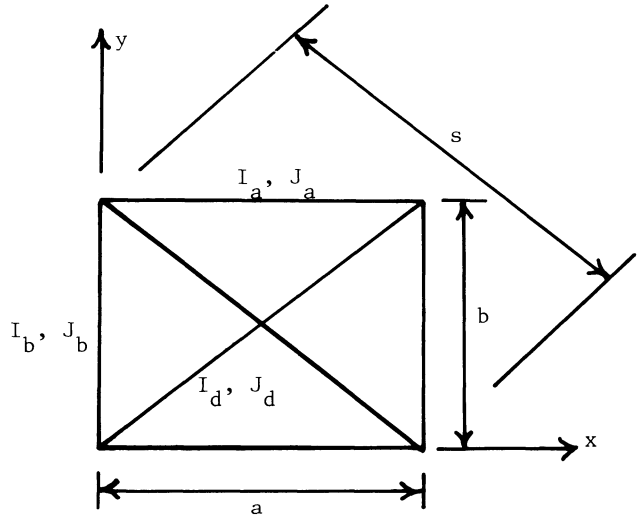


Figure 1. Grid-Framework Element

The matrix development, as outlined by Wang (1970), starts with the building of a statics matrix [A] relating internal forces and external forces, statically, and the building of a member stiffness matrix [S] relating internal forces and internal displacements through slope-deflection. Internal forces are classified as two bending moments at grid member ends and a member torsional moment. External forces or load points are two orthogonal xy moments and a verticle z force at the corners. Displacements, both external and internal, correspond in direction to their respective forces. Multiplication of the [A] and [S] matrices, according to Equation 1, produces the element stiffness matrix [k] of Table 1 tying external forces {P} to external displacements {X}.

$$\{P\} = [A][S][A]^T\{X\} = [k]\{X\} \qquad 1$$

The [k] matrix coefficients for individual elements are assigned to a structure stiffness matrix. When all coefficients have been assigned, Equation 1, in its global form, is solved simultaneously to obtain the {X} terms. A final solution is achieved by substitution of the appropriate {X} terms into Equation 2 to get the internal forces {F}.

Table 1. The [k] Matrix for the Grid-Framework Element

F/X	1	2	3	4	5	6	7	8	9	10	11	12
1	N+R+ 2D+2I	2H−Q	T+K	D	0	−T	I−R	H+Q	−K	−N	0	0
2		2A+M+ 2V+P	B+J	0	−M	0	H+Q	V−P	−J	0	A	−B
3	SYMMETRIC		C+F+L	T	0	−F	K	J	−L	0	B	−C
4				N+R+ 2D+2I	−2H+Q	−T−K	−N	0	0	I−R	−H−Q	K
5					2A+M+ 2V+P	B+J	0	A	−B	−H−Q	V−P	−J
6						C+F+L	0	B	−C	−K	J	−L
7							N+R+ 2D+2I	2H−Q	−T−K	D	0	T
8								2A+M+ 2V+P	−B−J	0	−M	0
9									C+F+L	−T	0	−F
10										N+R+ 2D+2I	−2H+Q	T+K
11											2A+M+ 2V+P	−B−J
12												C+F+L

$A = \frac{2}{a} EI_a \quad F = \frac{12}{b^3} EI_b$

$B = \frac{G}{a^2} EI_a \quad V = \frac{2a^2}{S^3} EI_d \quad K = \frac{6b}{S^3} EI_d$

$C = \frac{12}{a^3} EI_a \quad H = \frac{2ab}{S^3} EI_d \quad L = \frac{12}{S^3} EI_d \quad P = \frac{b^2}{S^3} GJ_d$

$D = \frac{2}{b} EI_b \quad I = \frac{2b^2}{S^3} EI_d \quad M = \frac{GJ_b}{b} \quad Q = \frac{ab}{S^3} GJ_d$

$T = \frac{G}{b^2} EI_b \quad J = \frac{6a}{S^3} EI_d \quad N = \frac{GJ_a}{a} \quad R = \frac{a^2}{S^3} GJ_d$

$$\{F\} = [S][A]^T\{X\} \qquad 2$$

Equation 1 is the key to finding the member properties I_a through J_d. Force-displacement relationships for a plate are deduced by moment-area. These relationships are substituted into Equation 1 to generate Equation 3.

$$I_a = \frac{(b^2-\mu a^2)I}{2b\ (1-\mu^2)} - \frac{ab^2+a^3}{s^3}\left[\frac{GJ_d}{E}\right]$$

$$I_b = \frac{(a^2-\mu b^2)I}{2a(1-\mu^2)} + \frac{b^3+a^2b}{s^3}\left[\frac{GJ_d}{E}\right] \qquad 3$$

$$I_d = \frac{S^3\mu I}{2ab(1-\mu^2)} + \frac{GJ_d}{E}$$

$$\frac{GJ_a}{E} = \frac{b(1-3\mu)I}{2(1-\mu^2)} - \frac{ab^2+a^3}{s^3}\left[\frac{GJ_d}{E}\right]$$

$$\frac{GJ_b}{E} = \frac{a(1-3\mu)I}{2(1-\mu^2)} - \frac{b^3+a^2b}{s^3}\left[\frac{GJ_d}{E}\right]$$

Thus, the member properties are dependent upon the modulus of elasticity, E, shear modulus, G, and moments of inertia per unit width, I, of the plate where $I = t^3/12$ and t is the plate thickness. All properties are tied to the torsional stiffness of the diagonals. When J_d is zero, the equations revert to the solution by Yettram and Husain (June 1965).

BRANDE IV Limit Analysis Procedure

There are three restrictions incorporated into a BRANDE IV limit analysis working procedure:

1. Only joint concentrated loads, both moment and force, are permissible.
2. The loading is proportional throughout an analysis.
3. Only a single load condition is allowed.

The third restriction is a concession to the added bookkeeping and greater time consumption rather than an absolute requirement.

The proportion and not the magnitude of loads is important in any cycle and so the smallest of the loads for the first cycle is generally of unit magnitude. For each elastic cycle, that

load magnitude sufficient to create a plastic hinge is determined by a load factor. The collapse load is the cumulative load factor from each cycle up to failure.

The analytical steps associated with a limit analysis of plates are:

1. The plate dimensions, node or joint coordinates, element locations within the nodes and, finally, the properties are defined. The structure is loaded with proportionate nodal forces and/or moments and analyzed as per Equation 1 and 2 to initiate the first cycle.
2. At the conclusion of the first cycle, load factors are computed as the ratio of the member plastic moment capacity to the member end-moments of the grid-framework elements. For each succeeding cycle, additional load factors are computed by dividing the remaining moment capacity by the member end-moments. If, at some member end, these moments have opposite signs, the additional load factor is computed by dividing the numerical sum of the plastic moment capacity and cumulative moment with the end moment numerical value from the current cycle.
3. For each cycle, all end-moments and joint displacements are multiplied by the smallest of the load factors. The end-moments and joint displacements are then added to their corresponding cumulative values from the previous cycles to get a new cumulative total. The formation of a new plastic hinge occurs when a cumulative internal end-moment reaches the plastic moment capacity for that member.
4. The location of the plastic hinge, therefore, is defined by the location of the smallest load factor. To account for round-off, additional hinges are established at those locations where cumulative moment falls within 0.01% of the plastic moment capacity. Load factor computations are not necessary at plastic hinge locations from previous cycles.
5. A new cycle starts with the re-building of the element stiffness matrices accounting for plastic hinges through matrix modification. Like the first cycle, the element stiffness coefficients are assigned to a global system stiffness matrix. The global matrix equation is solved simultaneously for displacements and the displacements are used to solve for end-moments returning to Step 2. The steps are repeated until collapse or twenty cycles, whichever comes first.

The collapse of a structure occurs when part or all of a structure fails. The collapse may be predicted by two events during the process of the limit analysis:

1. A singular system stiffness matrix or collapse is reached when there are sufficient hinges to cause a zero pivot during inversion.

2. Collapse occurs when there is a substantial % increase (greater than 1000%) in maximum cumulative displacement from the previous cycle.

An element would have to undergo inordinate or even impossible distortions for a member torsional moment in a grid-framework element to reach the plastic moment capacity. This is a strong indicator that something in the problem is incorrect. Thus, an error message is printed and the program terminated should this occur.

Matrix Manipulation-Plastic Moment Capacity

A grid-framework element is converted to elasto-plastic through the manipulation of the member stiffness matrix [S]. Force-displacement equations for a typical prismatic grid member with ends i and j are defined as:

$$F_i = \frac{4EI}{L} e_i + \frac{2EI}{L} e_j = S_{ii} e_i + S_{ij} e_j$$

$$F_j = \frac{2EI}{L} e_i + \frac{4EI}{L} e_j = S_{ji} e_i + S_{jj} e_j \qquad 4$$

The installation of a plastic hinge requires that additional moment at the hinge point be zero for the remainder of the analysis. Thus, F_i is set to zero, given a hinge at i for example, and Equation 4 is solved for F_i in terms of rotation e_j. The stiffness coefficients become $^{J}S_{ii} = S_{ij} = S_{ji} = 0$ and $S^{J}_{jj} = 3EI/L$.

The torsional stiffness coefficient is found through integration by plastic and non-plastic segments over the length of a member. The shear modulus in a totally yielded portion is different from that of a non-yielded portion. Formulas from the ASCE Guide to Plastic Design (1971) may be used to estimate the plastic hinge length and strain-hardened shear modulus of steel wide flange sections. For elastic behavior or when no reduction in torsional stiffness is assumed, the torsional stiffness is given by Equation 5.

$$F_k = \frac{GJ}{L} e_k = S_{kk} e_k \qquad 5$$

The matrix manipulation procedure is incorporated into BRANDE IV. A torsional reduction factor, ALPHA, defined as the ratio of the length of a plastic hinge to the total length may be user specified through the element properties data section.

By default, ALPHA is set to zero or the torsional stiffness is that of Equation 5. Other required properties are the strain hardened shear modulus, GST, and the yield stress of the material, YST, both input under the constants data section. Default values are 800 ksi and 36000 psi, respectively, typical for steel.

Plastic hinge locations are based on the plastic moment capacity of the grid framework members. The plastic moment per unit width of a plate is given as:

$$M_p = \frac{f_y t^2}{4} \qquad 6$$

where: f_y = yield stress of the plate

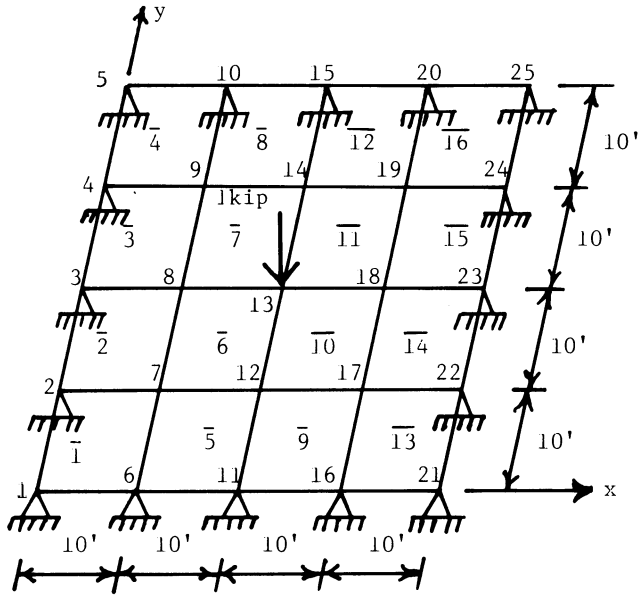
t = thickness of the plate.

For square elements, the grid-framework plastic moment capacity is equal to M_p times the element width divided by 3.41 which is established as the BRANDE IV default value. The static summation of grid member moments at a fully yielded central node divided by the element length reduces to the plastic moment of Equation 6 confirming the definition of plastic moment capacity as per above. The BRANDE IV user may specify a plastic moment capacity in the element properties data section.

Example Problem 1

The first of the example problems is a square simply supported steel plate with a concentrated central load as illustrated by Figure 2. The BRANDE IV data statements required to analyze this structure are presented in Table 2.

BRANDE IV data statements, modelled after ICES STRUDL II (1970), are designed to ease the work of specifying the structural dimensions, material, and element form. The TYPE statement designates the structural type, in this case, a plate. Other types are girder-diaphragm and plate girder-diaphragm structures. Three systems are possible, RIGHT, SKEW, and GENERAL, that define the shape of the structure. XD are the distances between columns of nodes and YD are the distances between rows of nodes. Nodal and element numbering always follow the pattern of Figure 2. Supports are established by the FIX command. Columns are numbered from left to right and rows from bottom to top. Thus, the first and last columns and rows of nodes are fixed against vertical or Z displacement. Elements 1 to 16 are parallelograms with a thickness of 0.25 ft. with material constants consistent with steel. The load is a vertical downward force of an initial unit magnitude acting on joint 13. Two analysis options are available, elastic and plastic, where the latter is a grid framework limit anlaysis.



1 = Joint Number $\quad\quad$ $\bar{1}$ = Element Number

Figure 2. Steel Plate of Example Problem 1

Table 2. BRANDE IV Data Statements

```
TYPE PLATE
UNITS KIPS FEET
SYSTEM RIGHT PLASTIC
XD  10.  10.  10.  10.
YD  10.  10.  10.  10.
FIX COLUMNS 1  5  Z
FIX ROWS  1  5  Z
ELEMENT PROPERTIES
1 TO 16 PARALLELOGRAM THICKNESS 0.25 ALPHA 0.
UNITS INCHES
CONSTANTS
E  30000.  ALL
G  12000.  ALL
POISSON  0.25  ALL
GST  800.  ALL
YST  36.  ALL
UNITS FEET
LOADING 1
JOINT LOADS
13  FORCE  Z  1.0
PLASTIC ANALYSIS
FINISH
```

Figure 3 presents the force-displacement (P-Δ) curve for joint 13. The load at first yield is 274.8 kips which continues to grow in ever decreasing increments up to failure at 514 kips. On the other axis, the displacement at first yield is 0.899 ft. with substantial yielding up to a failure displacement of 3.719 ft. The P-Δ curve conforms to elasto-plastic behavior exhibited in past analyses and interprets the tremendous reserve capacity that steel plate structures have in the plastic region. The plate had an 87% increase in load carrying capacity beyond first yield not adequately accounted for with an elastic design. One major benefit of a BRANDE IV limit analysis would be to present the designer with evidence of the reserve plastic capacity of plates permitting more efficient and cost effective designs.

A yield line analysis of the plate produced a failure load of $8M_p$, or 648 kips, where M_p is the plastic moment of 81 kip-ft/ft from Equation 6. This is in substantial excess of the limit analysis value of 514 kips. A likely reason for this discrepancy is revealed by the limit analysis yield pattern represented in successive stages in Figure 4. The concentrated load at the center of the plate should produce a high moment directly beneath the loads. Thus, initial yielding in all directions should occur at this point. As the load increases, the pattern should extend to progressive yielding about the diagonals. This is confirmed by Figure 4. However, yielding does not progress to the supports and, in fact, occurs about the first interior row of nodes. Failure must be termed premature collapse due to local yielding about the concentrated load. It should be noted that a limit analysis occurance of this nature is conservative or the yield line collapse load is not exceeded.

Example Problem 2

The second example problem is the three span continuous steel plate of Figure 5. Loading consists of unit forces acting at the central column of nodes. Collapse occurred after 13 cylces with a failure load of 54.1 kips at each central node and a maximum displacement prior to failure of 1.77 ft. A yield line analysis of the structure, modelled as a continuous beam, produced a collapse load of 54 kips. The limit analysis plastic flow of Figure 6 fits the predicted yield line pattern. Since both patterns are identical, the two failure loads must also be identical which is indeed the case.

Example problem 2 was repeated using a slightly modified structure where the length of the two end spans is the same as that of the center span. The collapse load is again 54 kips per central node but with a maximum displacement of 2.64 ft. Thus, the modified structure is absorbing more energy. This is reflected in the limit analysis yield pattern which revealed the formation of more plastic hinges (greater energy) with a wider scatter at the interior supports. The equality of the collapse

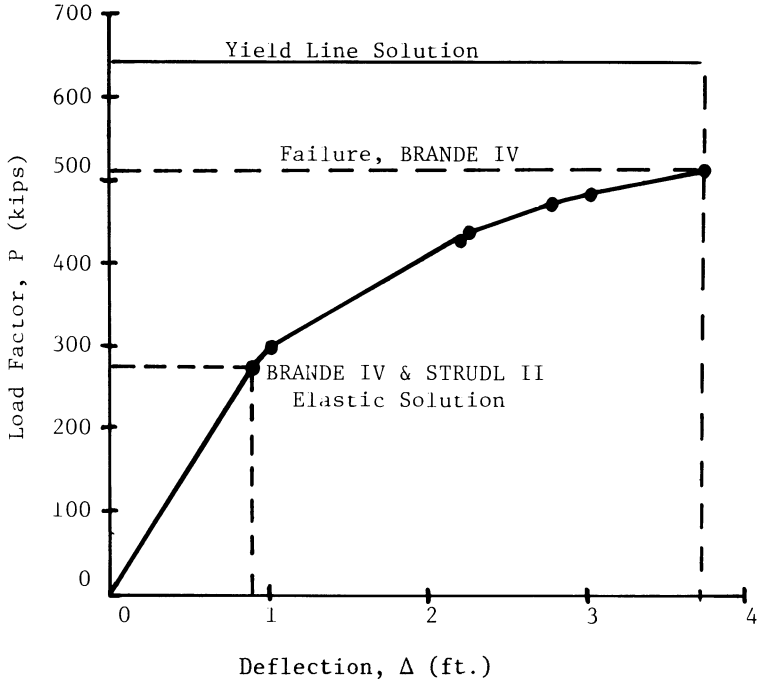


Figure 3. P-Δ Curve, Joint 13 of Example Problem 1

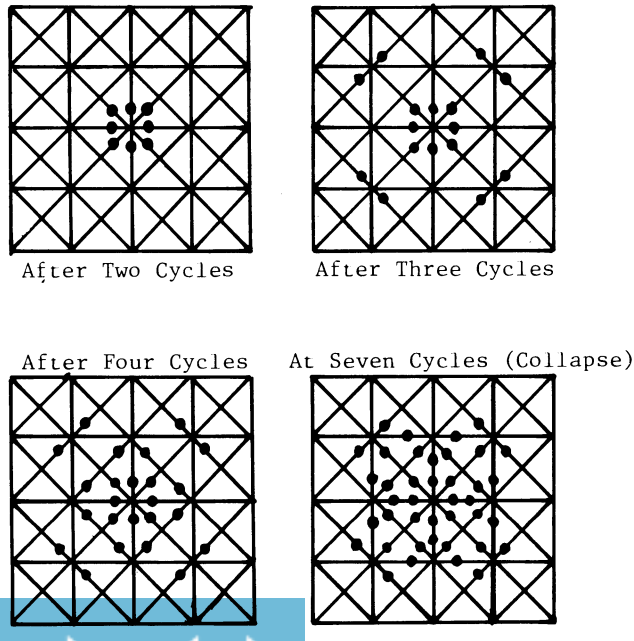


Figure 4. Plastic Hinge Formation, Example Problem 1

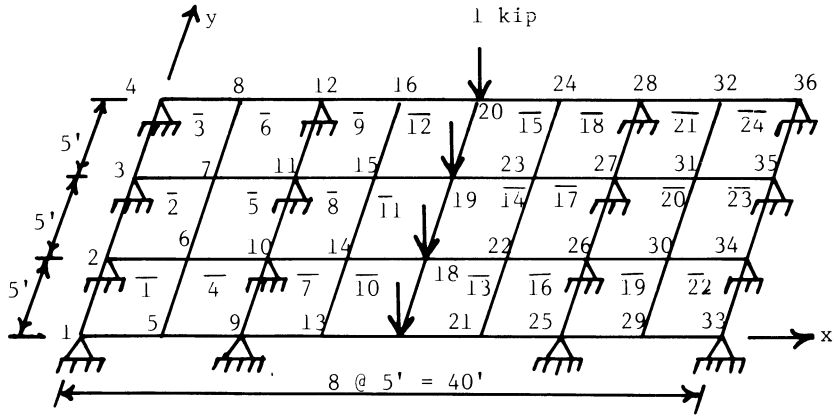


Figure 5. Continuous Steel Plate of Example Problem 2

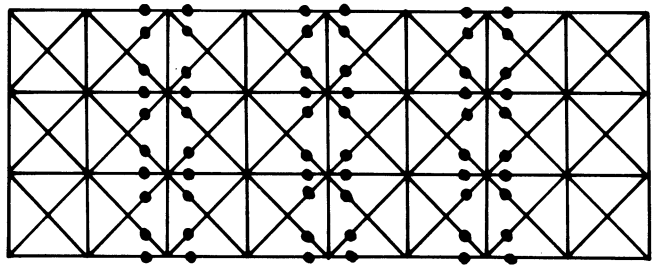


Figure 6. Plastic Hinge Collapse Mecanism, Example Problem 2

loads between the two limit analyses is predicted by yield line theory.

Conclusions

Two basic conclusions arise beyond the previously referred to advantages of the grid-framework BRANDE IV limit analysis. First, the collapse load is independent of the number of elements used provided that:

1. The nodal placement permits a reasonable path for plastic flow, i.e. the flow follows, as accurately as possible, the correct yield pattern. As a result, nodes grouped in areas of stress concentrations or in regions of plastic flow direction uncertainty will likely expedite the analysis.
2. The number of elements is sufficient to produce a reasonably accurate elastic finite element analysis.

Second, the plastic capacity of rectangular grid-framework elements has not as yet been firmly defined requiring additional study.

REFERENCES

American Society of Civil Engineers, Plastic Design on Steel - A Guide and Commentary (1971) 2nd. ed., ASCE - Manuals and Reports on Engineering Practice, No. 41, New York.

ICES STRUDL II - Engineering User's Manual, M.I.T. Department of Civil Engineering (1970) 2nd. Ed., Cambridge, Massachusetts.

McCarthy, W.C. (1982) An Elastoplastic Plate Bending Finite Element With Application to Bridge Structures, A Final Report to the Engineering Foundation, New York.

McCarthy, W.C., K. White, J. Minor (1980) BRANDE IV - A Bridge Analysis System, Proceedings of the 2nd Conference on Computing in Civil Engineering, ASCE, Baltimore, Maryland.

Traina, L. (1968) Matrix Analysis of Plate Structures in Bending, Doctorate Thesis, University of Wisconsin.

Wang, C.K. (1970) Matrix Methods of Structural Analysis, 2nd. ed., International Textbook Company, Pennsylvania.

Yettram, A.L. and H.M. Husain (June 1965) Grid-Framework Method for Plates in Flexure, Journal of the Eng. Mechanics Division, ASCE, Vol. 91, No. EM3.

AN APPLICATIONS SOFTWARE LIBRARY FOR FINITE ELEMENT ANALYSIS

I.M. Smith

University of Manchester

INTRODUCTION

Previous International Conferences on Engineering Software have included sessions on "Finite Element Systems". In this context, a "System" has conventionally meant a program, integral with which may be some subroutines. Over the last quarter century, many such systems or programs have been marketed, and it may be questioned whether anything "new" can be said about software for finite elements. The present paper suggests that it can, and offers a philosophy which differs radically from that on which stand-alone programs have traditionally been based. The paper begins by contrasting the new approach to applications software with the old, describes the origins and development of the philosophy and explores its implementation with particular reference to hardware advances such as "micro" computers and vector processors.

PACKAGE OR LIBRARY?

For want of an alternative description, the conventional finite element "system" will be called a "package". This usually consists of a large program containing tens or hundreds of thousands of (Fortran) statements. In some of these packages, reference is made to "libraries" of finite elements, ie. it is possible in the analysis to choose from a shelf of subroutines for various tasks: beam elements, plate elements and so on. This is not the writer's concept of a library which will be outlined later. The subroutines in typical packages are used in the form of appendages to the program. In extreme cases, the writer knows of subroutines which have no parameters at all and which pass information to and from the main program solely through COMMON blocks. If parameters are used, it is still typical to rely on a COMMON storage area as a means of passing variables. This type of subroutine is therefore chronically lacking in one of the main attributes of a good sub-program or

or procedure, namely portability. That is, the subroutine has no independent existence outside the program to which it is appended. If it is to be used in conjunction with a new program, detailed knowledge of the host program's COMMON block structure is required, and alterations to the COMMON area in the subroutine are likely. Thus the subroutine cannot be precompiled and stored away in what the writer understands by a user "library" by analogy with the permanent library accessible in all high level languages, and containing popular routines such as SIN, ATAN, ABS and so on.

In contrast, it is possible to write subroutines, even in an unpromising language such as Fortran, which are designed to pass all input and output via a parameter list. Such a subroutine can then be completely portable and can be shared by several programs and indeed by several users. There is a slight organisational overhead inherent in this calling tactic but it is negligible in comparison to the overall advantage in terms of total software effort. In Fortran it is unavoidable that the parameter list can become quite lengthy compared with the equivalent in a better language such as Algol.

There are several mathematical subroutine libraries of this type already on the market, such as the NAG, Harwell and IMSL libraries. The concept in the present paper is the extension of such mathematical libraries into a specific applications area - namely Finite Element analysis. It is anticipated that similar extensions could be made, adopting similar principles, in various other areas of engineering analysis such as boundary integral equations, finite differences etc and in quite different fields.

The nature of the extension of a typical mathematical subroutine library into an applications area is as follows. The basic mathematical library is essentially retained, and forms what is called the "Level 0" library. This contains the basic building blocks necessary for the construction of finite element analysis programs - quadrature, linear algebra in the form of matrix operations, equation solvers, eigenvalue solvers, ordinary differential equation solvers and so on. Taking as an example the NAG library, routines in Chapters D01, D02, F01, F02 and F04 are the most directly applicable. However, in the present application, two deviations from NAG philosophy have been introduced at Level 0. Firstly, mnemonic names have been used to give physical insight into the operations performed - MATADD adds matrices and so on. Secondly, many more routines exploiting matrix sparsity, eg. banding, have been added where none exist in the NAG original.

But if this were all that were done, it is doubtful whether users could readily construct their own programs for finite element applications. So what is attempted is to assist users by providing example programs which form a "Level 1" library, constructed from the building blocks at Level 0. Of course documentation problems then become quite severe, compared with the relatively simple provision of a set of test data and

answers at Level 0. The current Level 1 documentation (NAG, 1982) contains four introductory chapters describing the general principles of finite element analysis in the library context and thereafter each Level 1 program requires some 20 pages of detailed documentation, including a test problem, data and output which can be checked by the user before he embarks on a more ambitious analysis.

The case for the library philosophy has been made above in terms of portability of subroutines and consequent code sharing. However the philosophy is believed to have many advantages, offset by one disadvantage. The advantages are believed to include:

1. Reduction of duplication of effort at Level 0.
2. Ease of cooperation and exchange of software at both Levels.
3. Conciseness of Level 1 programs making for easy comprehension. A typical large program occupies a few A4 pages.
4. Level 1 programs are readily adaptable to new situations.
5. Only the necessary code for any application is loaded (see later in conjunction with "micro" computers).
6. New and better routines can readily be added/substituted at Level 0.
7. The educational advantages of modular, concise, simple to adapt programs are very considerable.

The major compensating disadvantage is that, unless in exceptional circumstances, there is no escape from the user understanding enough about the Level 1 coding to be able to modify it. A surprising number of users would rather spend 5 days preparing redundant data for a large "package" program than 1 day understanding how a Level 1 library program really works. Thus it may be that in the present climate, the greatest applications of the library philosophy programs are likely to be in teaching and in R and D departments of high technology companies as well of course as in academic research.

ORIGINS DEVELOPMENT AND FUTURE

The finite element applications library grew up as a by-product of pure research. At a time of much enthusiasm for "customer-client" arrangements and sponsored targets in University research it can be said that no one sponsored, funded or foresaw the need for such a software tool. Instead, experience of supervising many research and project students working on diverse applications of finite elements led the writer to evolve a system whereby each succeeding student could build at least partially on the foundations laid by his predecessors. In contrast, the writer knows of many research projects based on the "package" idea of software in which the student spends years understanding the coding of his predecessors and months on incorporating his own amendments before abandoning yet another monolithic package to be interpreted by his successors. In the library approach, the legacy of each research student is likely to consist of a few subroutines which are added to the

library. The next student begins with a Level 1 program of a fairly basic nature which is as close to his ultimate goal as possible, and may incorporate some of the bequeathed subroutines. Thus duplication of effort is kept to the minimum.

Another feature of historical interest may be that the library did not originate in Fortran. Indeed the writer was lucky enough never to have written in Fortran for the first fifteen years of his programming experience. Therefore the question of parameter passing, which occupied the present article earlier, never arose. In Algol 60, Atlas Autocode and Algol 68 - the environments in which the library was written - passing of arrays as parameters is easy and a natural feature of the language. It is an interesting speculation whether libraries would ever have developed at all had Fortran been the only coding medium.

In 1976 the SERC (Science and Engineering Research Council) set up its Interactive Computing Facility to satisfy the growing hardware and software needs of United Kingdom research workers. A user survey delineated two distinct classes of potential users of finite element software which they termed "users" and "developers". The "users" were supposed to have needs which could be catered for by packages although even these were essentially restricted to solid mechanics ("stressing") applications. The "developers" wanted to make significant departures from any available packages and the SERC decided that the writer's Finite Element Library represented the best solution to their needs. Considerable work was therefore undertaken by SERC staff to turn the Library from a private, inadequately documented and supported piece of software into a marketable product. A major improvement was made in the areas of pre- and post- processing. It now forms the software base for many research programmes throughout United Kingdom Universities and Polytechnics and has been quite widely distributed abroad.

Since the Level 0 library, although predating the NAG mathematical subroutine library, was so similar to it, collaboration with NAG seemed to be a natural progression and in 1982 the NAG organisation took over the distribution and marketing of the Finite Element Library in parallel with its original mathematical software library.

What has been done so far has been limited in scope by manpower restrictions. However future developments afford an excellent opportunity for improving and extending the Library at both levels.

MERGING WITH OTHER LIBRARIES

An indication of the scope for cooperation and exchange of software can be gained from the ease with which other Level 0 libraries can be substituted for the present one. The writer has experience of using Harwell routines for eigenvalue extraction using a Lanczos algorithm in place of the QR

algorithm originally incorporated in the NAG Level 0. Substitution of the new algorithm into a Level 1 program took a day and it seems that as long as the basic philosophy of parameter passing is adhered to, merging of libraries presents little difficulty.

MAINFRAME MINI OR MICRO

Even at the time of formation of the ICF by SERC, referred to above, it was anticipated that most Level 1 analysis runs would take place on a "mainframe", in their case an IBM 370/195. The interactive network, based on GEC 4000 series and Prime 400 "minicomputers" was merely expected to be used for debugging and pre/post processing. In the event, "minis" now prove powerful enough to do quite significant finite element analysis work. Furthermore, advances in technology mean that in future "micro" computers, possibly operating in single user mode, will become powerful enough for finite element work on a reasonable scale. The positive aspect of these developments is that the library is an ideal vehicle for micro software. The programs are concise, and only the parts of code which will be executed are loaded. The alternative philosophy adopted by many packages is to have very large codes, parts of which are skipped during execution. These would involve overlaying on a scale which is likely to tax micro operating systems.

The negative side of the micro implementations is the weakness of the present standard micro computer. Its possession of only 8-bit words means that double precision working has to be the standard. (This proved to be the major problem with minis where the 16-bit wordlength, lack of double precision complex and so on limited the transfer of some mainframe programs). However, when 16-bit working, or better 32-bit, is the standard on micros the library philosophy seems well adapted to exploit their power.

VECTOR PROCESSORS

A dilemma faced by all purveyors of engineering software, particularly for the larger mainframes, is whether to attempt to provide code which will exploit the power of vector processors. Under favourable circumstances, such vectorised code can achieve large savings in processing time. For example NAG are vectorising sections of the mathematical library and these subroutines could obviously be exploited by Level 1 programs in the Finite Element Library. However a cautionary note should be inserted. In the writer's opinion the central feature of the library system, to be preserved at any reasonable cost, is its modularity. Very often, to exploit vector processor power, convoluted algorithms are necessary which can destroy modularity. This should only be done as a last resort, because processor cost is becoming less and less critical compared with programmer cost (see below).

IN-CORE VERSUS OUT-OF-CORE

Some users of finite element programs cannot conceive of any mode of execution which is "in-core". Largely responsible for this situation is the well known and successful "frontal solution" technique popularised by Irons for example. This was originally devised so that equations with large band (front) widths could be processed on machines with very small core storage, and so considerable use was envisaged of transfers to and from backing store (disc). However, it makes little sense always to use this strategy on mainframes (CDC 7600 - 100K words of core, Cray I - 250K words, Cyber 205 - 2M words) or on virtual minis (eg. GEC, Prime) or on single user micros (eg. PERQ). Even if bandwidths are very large the writer uses a block skyline out of core solver to minimise disc transfers and it is known that for the same node numbering system, this will involve the same amount of arithmetic as a frontal solver. As with vectorisation, out of core versions of both levels of library are quite feasible if the demand is a continuing feature. Portability from machine to machine (already a major problem - see below) is much more difficult for out of core work.

USER EDUCATION

Since experience shows that, even when example programs are provided at Level 1, users can find adjustment to the library philosophy difficult, practice has been to run one-week courses introducing users to the system. Enough of these have now been run for it to be clear that many novel problems can be at least partially solved at the end of the week. The main barrier to be overcome is fear of altering the Level 1 test example.

CONCLUSIONS

Experiences with a finite element applications library have been described. It is believed that such a library does help towards resolving the "software crisis" wherein the cost of software escalates at a time of cheaper and cheaper hardware. Problems of portability from machine to machine still exist (at least 50 versions of the NAG Fortran mathematical library are necessary) but this is the responsibility of the implementer. The user sees complete portability of his code from program to program on the same machine.

Of course the very power provided by such libraries can be dangerous. Most finite element solutions are to problems of static equilibrium (steady state) wherein it can be said that numerical difficulties such as ill-conditioning are rare in physically well-posed problems. But substantial numerical problems still exist in the two other major types of finite element solution, namely eigenproblems and ordinary differential equations. In the former case some algorithms such as QR are

quite inefficient in many large problems whereas faster more approximate algorithms such as Lanczos can experience convergence problems, difficulty with multiple roots and so on. In the case of ordinary differential equation solution, choice of the appropriate method is still far from easy. Completely automatic methods such as the Gear algorithms are often quite inappropriate.

BIBLIOGRAPHY

Du Croz, J. (1982) Programming Languages for Numerical Subroutine Libraries. NAG Newsletter No.1, May.

Ford, B. (1978) Parameterization of the Environment for Transportable Numerical Software, ACM Trans Math. Software, 4, 100-103.

Hague, S.J. and Ford, B. (1976) Portability - Prediction and Correction, Software - Practice and Experience, 6, 61-69.

NAG (1982) Finite Element Library, Levels 0 and 1. Documentation by C. Greenough and K. Robinson.

Smith, I.M. (1982) Programming the Finite Element Method, John Wiley, Chichester.

AN INTERACIVE APPROACH TO CONTROL NONLINEAR FINITE ELEMENT SOLUTION PROCEDURES

Th. Friedrich

ETH Zurich

INTRODUCTION

The application of linear finite element programs is wide spread and it seems to be a everyday tool of the structural engineer. Nowadays nonlinear finite element analysis gains more and more interest as engineers want to study the real behaviour of a structure within the serviceablity and ultimate state. Tests specimen have told us that structural members do behave nonlinear very early under small load intensities. Especially certain material laws demonstrate a significant nonlinear stress-strain relation in short time but also in long term loading. Only nonlinear analysis can take into account the geometrical and material nonlinearities. Although the nonlinear finite element analysis is a quite new developped branch and most methods are still under investigation, some programs are still used for practical design. But most users aren´t satisfied with the application, because they spend a lot of time and computer-costs without receiving reasonable results. In most cases procedures don´t converge or they produce results, which aren´t compatible with the real structural behaviour.
As the nonlinear analysis process is an incremental and iterative one it will need parameters to steer its successful termination. Most programs require data from the user at the very beginning for a calculation run, from which he doesn´t know how it will behave. The existing programs try to surmount this problems by fitting automatically the parameters according to the actual internal state. But as there is a great lack of these automatic procedures, the most programs aren´t so successful to provide reasonable results.

Our approach try to involve the user with the calculation process, so that he can interact with the program and change the parameters according to the current state. The computer will provide him with information and the user will decide what he is going to do. By this idea the user gets more familiar with the nonlinear behaviour and he can understand quite more, how a structure works within the nonlinear range.

REALIZATION OF THE FINITE ELEMENT PROGRAM

In the following we will describe a program, which has been developed as a part of a larger program system. The main project is called the FLOWERS system [1], which has been conducted at the Swiss Federal Institute of Technology in Zurich. This a module oriented software system for finite element applications.
As any kind of structural and nonstructural type can be analysed with these modules, they have to serve for all possible elements. There exists a element library out of which the different element types can be taken of. In order to allow the program to be connected with different element types, the modules have to be planned and implemented with quite exactly defined interfaces from the module to the element routines. When we analyse, what tasks the modules have to fulfil and how the element procedures have to contribute, we can describe the interface quite clear. If we think of nonlinear applications, the following data flow from and to element routines must be possible:

- elements receive local displacements and saved element data from the module

- the element routines build as a function of the actual displacement and the saved data
  - the local stiffnes matrix
  - the equilibrium loads
  - update element data

  and return them back to the module.

Having in mind this data transfer the software for general purpose finite element applications is possible to realize. Once a module works for a certain application, we can build up an extensive element library to count for different structural and nonstructural modelling. All FLOWERS modules have been implemented after this concept. FLOWERS possesses several modules for different tasks. For the linear application they all work in batch mode.

Input is given by a so-called syntax-diagrams, which makes the using very easy, because those diagrams are quite self-explanatory. The program I am going to describe, will treat the nonlinear static range with extension to time dependent material laws (creep,relaxation,shrinkage). It uses two other FLOWERS modules, one called ´SYSIN´ to describe the geometry, load data and the element typ properties and the other one called ´GRAPH´ for presenting saved results in a graphical form.
The program has been implemented on the DEC10 computer at the ETH-Zurich. It is written in standard Fortran and can easily transferred to any other computer type.

THE NONLINEAR ANALYSIS PROCEDURE

In order to understand how we can influence the calculation process let´s analyse the nonlinear procedure. In contrast to the linear solution, we can´t receive a final result within one single step, because the relation between load and displacement is no longer linear. Previously received results will influence the behaviour of the next solution. The load must therefore be incremented step by step onto the structural model. Even within these limited steps we don´t succeed in finding a direct solution. As we are obliged to linearize the equations, we inevitablely receive a mistake, which can be removed by further iterations. In order to receive an exceptable result, we need to define some convergence criteria. Different procedures (initial strain method, Newton-Raphson, modified Newton-Raphson) are well known to tackle the nonlinear analysis [2].

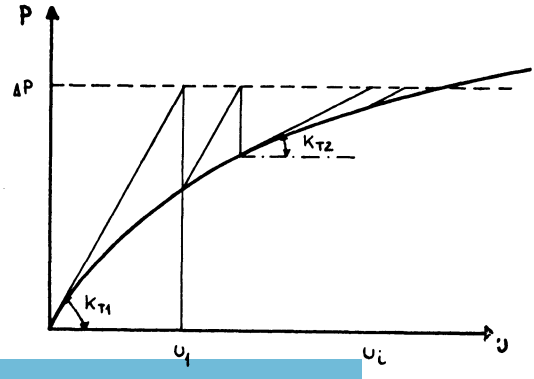


Figure 1

From Figure 1 we can see, that the reforming of the

stiffness matrix within an iteration cycle will speed up the convergence. We have seen, that there will be three different possibilities to set parameters and influence the nonlinear procedure:

a) step size (load and time)
b) convergence tolerance criteria
c) stiffness updating strategy

The parameters to be needed are dependend on the actual state of the analysis. To steer the calculation we need the interaction between the user and the process. There is a need for the input of the parameters, and there is another need to get information of the current internal state, in order to select the right parameter values.

COMMUNICATION BETWEEN THE USER AND THE NONLINEAR PROCESS

Input is done via menu list (see Figure 2).

(I) - INCREMENTATION

(C) - CONVERGENCE

(M) - MATRIX STRATEGY

(Q) - QUIT

Figure 2

The user take his choice out of different commands written onto the terminal by typing a character. This will lead him either to another menu or to a request to input the needed parameters.

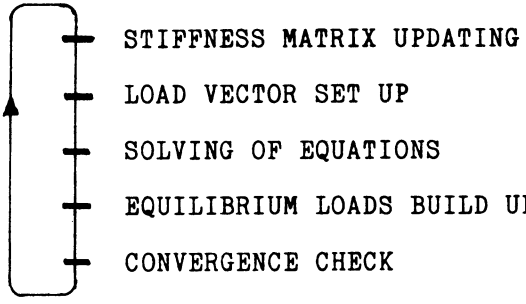


Figure 3

Information on the current state can be delivered to the user on various ways. During the process the actual kind of action is indicated on the terminal. The repeating actions of the iteration process are shown in the Figure 3. After being built, convergence check results are transmitted to the user and compared with the required ones. Other useful data, such as step number, iteration cycle, CPU-time etc. are also currently presented.
An other kind of getting deeper inside into the internal state is the data from the element and their belonging material law procedures. Different kind of messages can be transmitted. The user can define identifiers, representing various element data, which will be transmitted to the terminal during iteration. Element routines themselves will inform automatically, when a certain stress-strain level is reached (e.g. first plastification in an integration point). Whereas these messages will be treated as nonfatal ones, others can cause a execution break (e.g. maximum allowed strain reached). Beside the current output onto the terminal during the iteration cycles, the user can get more information upon the element we he stops the execution and asks for more data for each element, with will be transmitted onto screen and print output device.
Data can also be transferred from the user directly to the element routines. This means, that predefined element data can be changed during calculation. (e.g. activate or desactivate parts of the element, change material law properties, element behaviour should be linear or nonlinear). This is done by an identifier, which represents the predefined data.
The strategy how the complete nonlinear analysis will be done depends only on the user´s decision. The procedure can be executed single stepwize, pausing after each step or even after each iteration cycle. On the other hand some steps can be predefined and the program do not stop at each interval. An automatic incrementation up to the limit load can also be choosen. Once decide to increment several steps, the user can still interrupt by pressing any key and activating the ´keypress´ function, which will stop execution. Then again detailed information on the internal state can be requested and parameters can be inputted to steer the following calculation.

Some other helpful functions are available by the program. Different kind of output levels cause the program and the element routines to send output in a

sparse or extensive form onto the printer. Global or element data can be saved at defined levels, in order to use it for further application, such as graphical presentation or other postprocessing. Restart points can be set, and reloaded from. This allows the user to stop the execution while having a deeper look into the printed data and continue from this point at a later time. From a saved point further runs can be done with various strategies in order to compare results and accuracy with each other. The late aspect will help us much to gain more inside into nonlinear calculation. The nonlinear execution scheme is displayed in Figure 4. Besides the calculation oriented actions, we can see the possibilities when the user is informed, and how he can input the parameters.

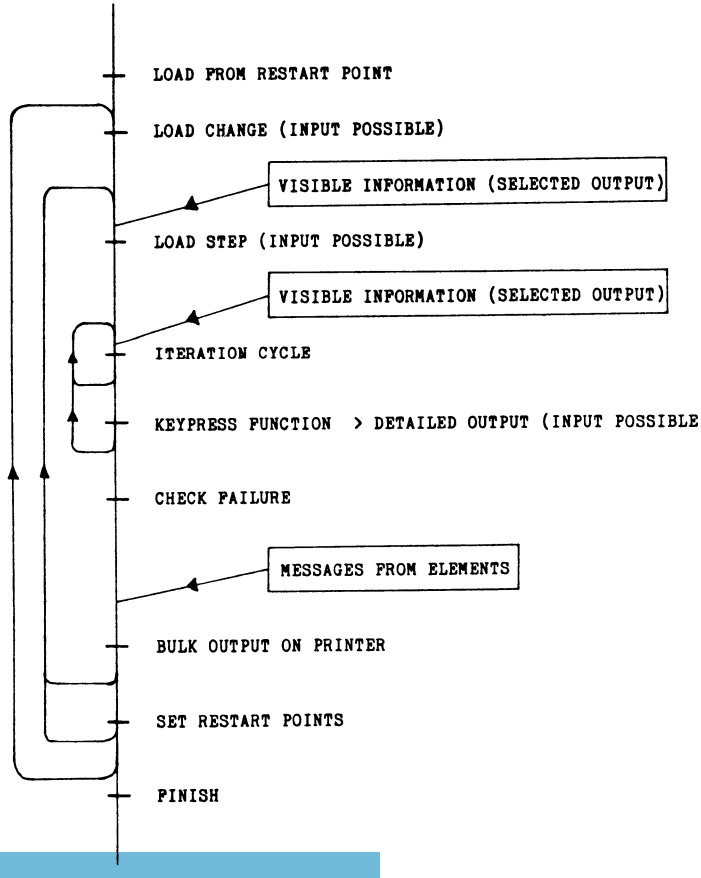


Figure 4

At predefined points, such as step end or during an automatic incrementation or iteration by activating the ´keypress´ function the user can change the parameters in order to influence the nonlinear analysis. He will take one or several of the above mentioned measurements (change step size, change stiffness matrix, release convergence criteria) to achieve a practical result.

EXAMPLE

We had to analysis a series of concrete columns under imposed deformation. We were interested in the results, how the limit load would change, if a restraint force had been obligate to the columns [3]. Two different failure mechanism can develop, depending on the ratio of the slenderness of the structure. If slenderness is high, we receive a stability failure, whereas small slenderness causes material failure, when strains are reaching the maximum. In any case we are also interested to control the stress level within the reinforced bars, in order to check the serviceability. The imposed deformation is first loaded onto the structure and the vertical load is then incremented up to the limit point. During step-by-step incrementation and during iteration cycle we did control first displacement and internal forces within selected nodal points and second some element data from various elements.

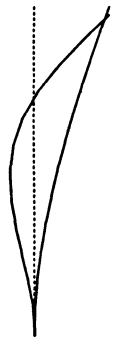


Figure 5

We could exactly follow how stress strain develop within the elements. Saved data was used to demonstrate graphically the results obtained by the step-by-step procedure. Figure 5 shows three different deformation shapes, undeformed, deformed under the imposed load and the deformation under limit load. Figure 6 shows the stresses on top and on bottom of the cross section, situated at the bottom of the column.

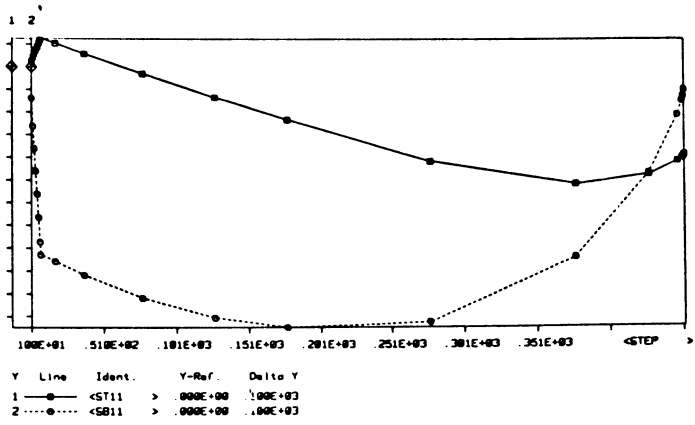


Figure 6

CONCLUSION

The interactive approach to control nonlinear finite element solution procedures will involve the user much more into the calculation process. It will help him to understand more on the nonlinear behaviour, but on the other hand he needs a lot of knowledge to interprete the results and input the right parameters. Advanced educated users are required, more than the black-box users. But once being familiar with such a program you can receive good results within a reasonable time, as you don´t do a hundred of runs, which all haven t converged and you have to reject the results. Needless to say that the size of the analysing model is limited by the computer power, because you will need enough power to solve your problem within a suitable time. But as computers are getting bigger, faster and even cheaper, we will have no problems in this direction.

The proposed and implemented approach will be a valuable instrument to analyse the nonlinear behaviour of structures. Reasonable results can be obtained and the user will gain a lot of understanding for the nonlinear calculation.

ACKNOWLEDGEMENT

I thank the members of the FLOWERS team, having me provided with program parts which I use in connecting with the decribed program.

REFERENCES

[1] E.Anderheggen, G.Bazzi, H.Elmer, T.Friedrich, H.Maag, J.Theiler: FLOWERS User´s Manual, First Edition June 1981, Swiss Federal Institute of Technology Zurich.

[2] K.J. Bathe: Finite Element Procedures in Engineering Analysis, 1982

[3] T. Friedrich, C. Menn: Behaviour of concrete columns under imposed deformation, to be published.

DESIGN BY ANALYSIS AND THE FINITE ELEMENT SYSTEM FINEL

Dennis Hitchings

Imperial College of Science and Technology

Oliver Hopkin

Babcock Power Limited

INTRODUCTION

The validation of components for nuclear power stations, from the standpoint of strength and safety, requires a detailed assessment, based upon stress analysis, of postulated modes of failure. The framework for such an assessment is given in Section III of the ASME Boiler and Pressure Vessel Code, which contains rules for the design against catastrophic rupture or collapse under a single application of load and against the effects of cyclic loading, leading to failure by fatigue or thermal ratchetting. For present purposes 'design' is taken to mean the establishment of component scantlings which conform to stated limitations on acceptable proportions and shape, and which satisfy stress based criteria depending upon the component and the design or operating conditions given in the design specification.

There are two basic approaches to carrying out the design activities, Design by Rule and Design by Analysis. Their fitment in Contract Design Activities is shown in Fig. 1. The former is the procedure required by codes such as BS 1113. It is based upon evaluation of pressure with a conservative allowable stress. It is carried out by substitution into formulae, the use of geometric rules and specified design conditions. No specific account is taken of cyclic operations or thermal stresses. Design by Analysis requires the calculation of temperature and stresses at each point of a component at a number of time points during each operational event, both planned and unplanned. All this data has to be taken into account in order to validate the component against cyclic loading. This has led to the evolution and application of powerful computer programs and systems, in order that the necessary calculations may be performed so that the large amounts of information generated in the design process can be handled. The present situation is that the great majority of such calculations are carried out by the finite element method.

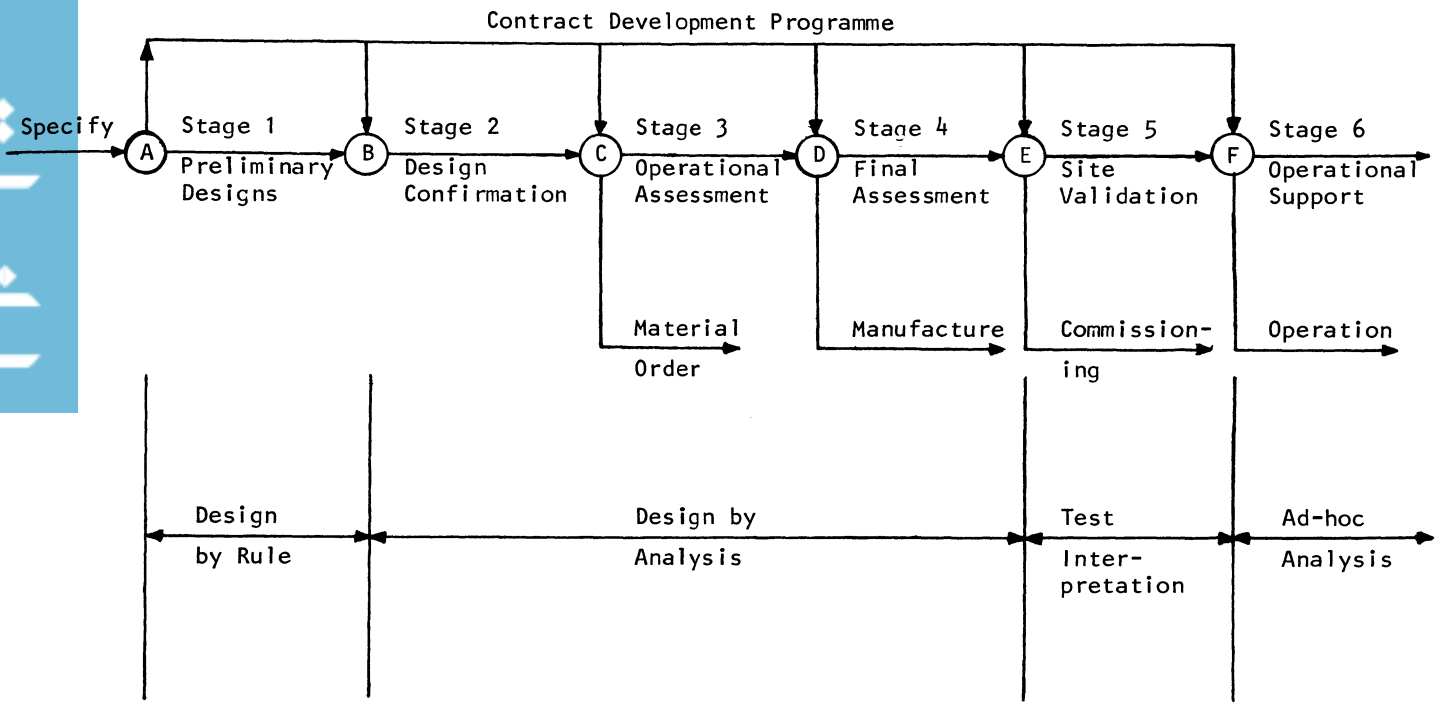


Figure 1 - CONTRACT DESIGN ACTIVITIES

CURRENT STATE OF THE FINITE ELEMENT METHOD

Many users now have considerable expertise in using the finite element method and they have become more demanding in their requirements for such programs to solve progressively more complicated problems as a matter of routine. There is currently a wide range of computer programs available for performing finite element analysis and, for the most part, they contain a range of overlapping facilities and produce identical results. From a users point of view it has been found that, for any given problem, there will be a best choice of program, since each has its own strength and weaknesses. In this respect it is unlikely that, in the near future, there will be any program that will contain all of the facilities required. This is because the required analysis may consist not only of standard design analysis requirements that are currently used, but will also require special 'rogue' types of analysis such as plastic analysis, creep, limit load, large displacements, buckling, fluid structure interaction and dynamic analysis with various types of excitation and damping.

As one might expect there is a great deal of uniformity in the type of data required for all finite element programs. However, each program has its own input format and these differ widely and require considerable experience on behalf of the users. Acceptance of the method has shifted the emphasis from refinement of the method itself towards more friendly access via pre- and post-processors. This has coincided with the emergence of interactive graphics as a practicable design office tool.

Three definite stages can be identified in a finite element analysis:-

1) Problem specification (usually interpreted as mesh generation but this is only one aspect of it).
2) The problem solution or calculation of the response parameters.
3) Results manipulation and assessment.

This last item involves the reduction of the output produced to more manageable proportions, often requiring a considerable effort by the engineer to interrogate and interpret the results into a more meaningful form. This is a surprisingly expensive procedure, often costing more than the problem solution itself. Additional costs are incurred in reformatting data to pass from one program to another.

NEED FOR A HANDLING SYSTEM

A more detailed investigation of this process has highlighted notable deficiencies, one of the most important of which is the inability to mix various idealisations within the current

handling systems. Such a feature is necessary for two reasons. Firstly as a design develops it changes and, for quality assurance (QA) purposes, this historic data must be recorded. Secondly, different analyses may require different idealisations, for example a mesh for fatigue analysis involving stress concentrations is unlikely to be suitable for dynamic analysis.

Many users see the main problem in using finite elements as the specification of the mesh and once this is generated the computer will do the rest. Unfortunately this has led many engineers to think that once the geometry is correctly specified the mesh is good and will produce accurate results. Even some experienced users only pay lip service to the fact that often the accurate representation of the overall geometry is less important than the detail design of the mesh in critical regions. The fact that the bulk of the input data is required to define the mesh has led to the current development of pre- and post-processors. If the problem were merely one of specifying data and performing simple manipulation of results then this would be acceptable. However, the volume and range of analyses that will be performed, together with stringent QA requirements means that none of these processors will be useful for more than a limited period in their current form. The problem of handling the vast amount of data associated with the method and the processors becomes as important as the technical problem itself. Experience has also shown that there is a need for the automatic production of reports, tables and figures from the computer, without any transcription by hand.

It is not possible in a design organisation to support more than two or three packages in-house, hence there is a need to link different types of finite element system and pre- and post-processors by means of data formats that are independent of the analysis program, since output files of any one program will not, in general, suit the requirements of the next. This process will also require a standard analysis project data base. All of these considerations point to the need for a simple handling system which will enable any processor to be accessed through a common input. The concept is now one of data flow rather than program flow.

GENERAL SPECIFICATION FOR THE DATA HANDLING TOOL

A reconsideration of the Design by Analysis process, in the light of previous statements, shows that it has four distinct phases:-

1) Problem definition and modelling.
2) Numerical solution.
3) Results manipulation and assessment.
4) Recording and reporting.

Some general control of the data handling process is needed, since each stage is inter-related with the others. Further rationalisation of the analysis process will lead to further efficiencies by splitting it into discrete steps of specific analytical tasks, each performed within an independently executable module. These are linked together by the computer with streams of data flowing betweenthem via a common project data base.

The use of such a rationalised system should allow for coupling with any analysis program that is to be used for component validation, with a smooth feedback from assessment operations to pre-processing. This will enable conclusions to be efficiently compiled from the analysis, including the modification of data for subsequent computations. Seen in this light the finite element method is only a part of this more general analysis tool. If the tool is modular and open ended in that it contains facilities for the incorporation of user subroutines, modules or analysis facilities, then it becomes applicable to any type of analysis.

The finite element system FINEL has a suitable architecture for development into this more general analysis tool, based upon its truely modular structure, its general internal data base, general input format suitable for describing any problem and its advanced graphical facilities.

THE FINITE ELEMENT PROGRAM FINEL

FINEL has been developed at Imperial College of Science and Technology, University of London in conjunction with Babcock Power Limited, London (BPL) to handle all aspects of the finite element analysis for both the R&D analyst who requires a flexible program on which to develop his ideas and for the engineer engaged on production work, who requires a proven system. FINEL is a highly structured program, the design of which follows directly the steps of the finite element method. These can be broken down into a set of discrete, almost unrelated, steps such as mesh generation, assembly, solution etc. This feature is used within FINEL to define a modular structure for the program. Each module defines a discrete aspect of the finite element method and can be considered conceptually as a super subroutine.

The system has a central executive which remains resident in core. The main prupose of this is to control the sequence in which the modules are executed, and hence to control the analysis that FINEL carries out. No module directly communicates with another, they all pass through the executive.

The modular structure is carried down one lower level within FINEL, by the definition of a series of FINEL libraries. Again this idea stems naturally from the finite element method

itself. Within the finite element analysis the continuum is divided into discrete regions, or elements. These elements can take different forms depending upon the actual problem being solved. Within FINEL the various types of elements that are available are gathered together in the FINEL Element Library. There is a variety of other libraries within the system, these being Analysis, Load, Plot, Module, Regions etc. Each library has a library executive program associated with it and this executive can be called by any module within FINEL. The program has been designed such that an engineer can add modules that can be called by the executive, or add routines to any library such that they can be called and executed without the system having to be modified in any way. All of the libraries have temporary entries associated with them that need be included only for one run. Any temporary additions are given the name OWN1, OWN2 or OWN3. This naming convention means that such temporary additions are identified on the output, allowing their use to be incorporated into any QA procedures. The temporary routines can migrate to a permanent status, with their correct name, when they have been fully verified.

As stated earlier, all communication is done via the Executive and this applies also to the data communication within FINEL. However, any large arrays of data items are not returned to the Executive and instead they are written to the FINEL Data Base in order for them to be communicated to other modules. The data base is a random access mass storage file which allows free access to any data that has been generated by a run of FINEL. It is organised as a two level system containing 30 files each with at least 500 pages, where a page can be of any length. The file is organised such that any page of any file can be accessed directly. Some files are dedicated to storing particular sets of data, whilst others are free for temporary use, as desired. For example File 1 is used for storing element data and any information relating to a given element can be found on the relevant page of File 1.

To perform an analysis the user calls a sequence of modules to execute the problem. For the convenience of production use standard module sequences are available within FINEL, for example the single statement ANALYSIS STRESS sets up the basic stress analysis sequence, as shown in Fig 2. However, merely following a module sequence from the start to finish and then ending the analysis can, at times, be too restrictive. To cater for all this an editting facility is provided within the executive to delete or insert modules as required. It follows that, as each module is executed, then it expects to receive the input data associated with it. The basic input data sequence is thus defined by the module sequence, which itself is the logical progress of the analysis. The very generality of FINEL itself and the wide variety of problems that it can solve calls for a non-specific form of the input

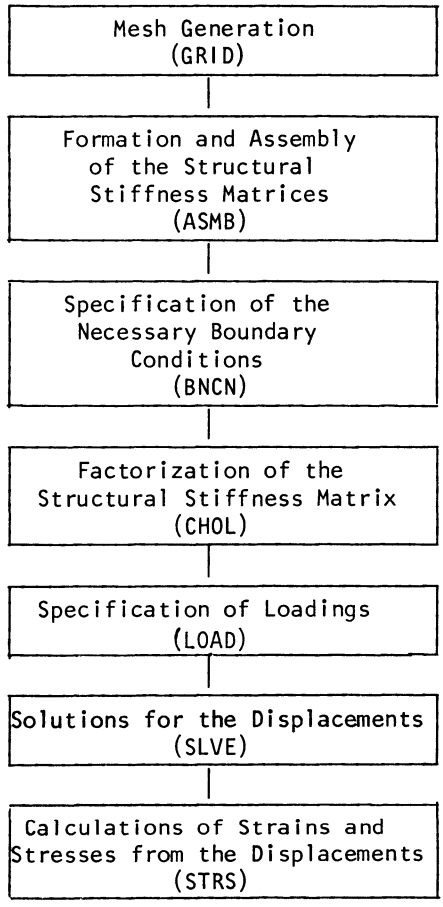


Figure 2 MODULE SEQUENCE FOR ANALYSIS STRESS

data. The smallest complete segment of information within the input is called a FINEL Data Block, each of which takes the same form, consisting of a command, a qualifying label and the actual data. The commands define the data block and the label qualifies the command. The data consists of the actual data and is usually numerical. The data input is in free format, designed for interactive use. The command/label form makes the input very easy to construct and check.

FINEL contains a pre-processor which is executed before the main program. This is done automatically and checks the syntax of each line. It also allocates central memory, such that it is used as efficiently as possible, from an inter-ogation of the input data, the analysis required and the elements used.

The system has also been designed to be user friendly in the real world of practising engineers. This is typified by the restart and sub-structuring facilities which allow the user to save and re-use the FINEL Data Base. For most computers this is achieved by standard commands within the FINEL input, with no change to the job control language. The form of the FINEL input and the restart capability is illustrated in Figures 3 and 4. This is the actual form of the input, apart from the fact that the numerical data has been replaced by comments

```
COMMAND     LABEL          COMMENTS

ANALYSIS    STRESS         ;Define a stress run
DEFINE      name 1         ;Define a name for the restart file
ELEMENT     name           ;Choose an element
MATERIAL    STIFFNESS      ;Define material properties
REGIONAL    COORDINATE     ;Define sufficient points to specify
                            the geometry
REGION      name           ;Automatic mesh generation
FIXED       FREEDOMS       ;Specify boundary conditions
LOAD        name           ;Specify loadings
END         JOB            ;End the analysis
```

Figure 3 - Typical FINEL Input Data

```
COMMAND     LABEL          COMMENTS

RESTART     name 1         ;Restart the analysis
RE-ENTER    LOAD           ;Enter the restart at the LOAD module
LOAD        name           ;Specify new loadcase
END         JOB            ;End the analysis
```

Figure 4 - Typical FINEL Restart Data

DATA HANDLING SYSTEM FOR DESIGN BY ANALYSIS

The modular structure of FINEL and its internal data base

allow for it to be developed into a general purpose handling program for all phases of an analysis. The current state of FINEL is reasonably representative of most well accepted FE programs in that it can analyse a specific mesh that is presented to it. The generation and handling programs associated with the FE analysis are also closely related to a specific FE mesh. This is very restrictive in the general philosophy of Design by Analysis, in that when a complicated structure subject to a wide variety of loads is to be analysed, it is unlikely that a single FE model will be sufficient. Instead a variety of models must be built. This presents two major problems, the first being to ensure that the same geometry is used in each model and, secondly, the results from each model must be integrated and assessed. There is also a requirement to interface with CAD/CAM programs to define the geometry and reporting programs to present the final results reports.

It is unlikely that a single data base will be sufficiently flexible and efficient to be used for all aspects of the design. Instead it is envisioned that each program will have its own local data base, but this is constructed from data controlled by a central global data base. The common information between all of the design aspects is the basic structural geometry and materials and this will be held on the global data base. This must also hold references to the information that is contained on the local data base.

For the analysis phase of the design it is envisioned that there will be two levels of data base. Each analysis program contains its own working data base and these will form the second level. The primary level of analysis data base is of more interest and is described here. There are three different geometry definitions

1) The physical geometry as derived from the global data base.
2) The FE analysis mesh.
3) A series of results surfaces.

The physical geometry provides a very detailed geometrical specification of the structure. It is not related to any type of analysis and will probably be much too detailed for any given analysis. It is termed the non-specific geometry since it is not related to any particular analysis. It need not be built by the engineer and is probably more logically set up by the Drawing Office on a CAD system such as CADAM.

The FE analysis mesh can be generated from the non-specific geometry as required. Great versatility is provided here, to approximate the true structural geometry for the purposes of the analysis. For example, typical BPL structures are almost axisymmetric mesh from a non-axisymmetric geometry. Further, the handling program can generate shell and axisymmetric thin

shell models from a three dimensional solid non-specific geometry. It will also be possible to mesh any part of the non-specific geometry, rather than the complete structure. Any actual FE mesh is not saved on the data base, instead a reference is saved to specify what has been done.

To assess the results in a meaningful way each analysis must be stored. However, the generality provided by the non-specific geometry is lost if the results from the FE mesh have to be saved directly, since each mesh will have results at different points in the structure. To overcome this the user can define results surfaces from the non-specific geometry. A results surface can either be a true surface or any surface within the body. The results from each FE analysis are then interpolated onto the results surfaces, before they are manipulated. In this way there is no incompatability between the various FE meshes and all of the information to hand can be used. The results surface concept is also very convenient for giving the engineer an easy way to manipulate the results for assessment. He can choose to look at only one, or a small set of, results surfaces at any one time. Since the results are grouped into surfaces already the interactive response is quick and incurs little machine cost.

The results surface concept is also applicable for moving from one analysis type to another. Typically within BPL a thermal conduction analysis is carried out and the temperatures found from this are fed into the stress run. Each analysis may have different meshes, either because different programs were used or because the analyses were conducted by two different groups. However, with the results on the analysis data base being located on the results surfaces, which are independent of the meshes, the translation of the results from one run to another is automatic. Note also that the concept of a non-specific geometry and results surfaces imply that the solution scheme does not have to be the FEM. Any solution process, such as finite differences or boundary element methods can be encompassed, even the results from hand calculations can be included! The analysis handler will be used for general archiving of any analysis results for the component.

The analysis handling program is to be based heavily upon FINEL and the experience gained in writing it. FINEL is used partly because its modular architecture and data base can be incorporated in the system, but more importantly, because it provides a direct link to the FE process. In many cases FINEL will be used for the actual analysis, allowing the transfer of information between the analysis data base and the FE data base to be very efficient. Currently, with the existing process of separate mesh generators, FE programs and results processors, the various data translation times and costs can be as great as the FE analysis costs.

With the analysis handling program being based upon FINEL it will be possible to generate equivalent loads, if these are not readily available in the package to be used. It will also allow stresses to be recovered from the displacement results from the FE analysis. This can overcome one of the major problems where each FE package outputs stresses at different points or coordinate systems. In fact, only the displacements need be passed back to the analysis handling program, stress recovery being considered to be a post-processing operation. For linear problems this can reduce the data transfer very significantly.

Finally, the analysis data base is being designed to be archivable, that is it can be incorporated into the QA process and can be used to reconstruct the history of the analysis of the component at any stage of its development. This is a special problem for companies such as BPL. where for large projects such as nuclear power stations design validation calculations may extend over several years, but it should also be a goal for every company that uses finite elements seriously.

CONCLUSIONS

The Design by Analysis process required detailed assessment of postulated modes of failure. This means that large amounts of data are handled to validate components. The acceptable tool for stress analysis is the finite element method. A detailed investigation of the process has indicated that there is a need for a simple handling system with the emphasis on data flow rather than program flow, and rigorous QA features. Without such a system it is considered that the current power of numerical analysis systems cannot be exploited in full. It is suggested here that no existing package contains all the currently required features and any such program would have a limited lifetime, since requirements will be subject to continuous change.

The finite element system FINEL has a suitable format for developing into this more general analysis tool, based on its modular structure, general internal data base and the ease with which user modules can be incorporated. Its basic facilities have been described, as have developments for data handling with particular emphasis on its capabilities in terms of general data and results manipulation techniques that are independent of the analysis method used. FINEL is in use at BPL for extensive computer analyses in the fields of thermal, stress, fatigue and fracture analysis in support of supply contracts and as a specialist service direct to clients. FINEL is used to work at the forefront of a number of related technologies including analysis of dynamic response, seismic analysis and impact.

Whilst concentration has been upon nuclear requirements and the finite element method in particular, the concept is completely general and can be applied to any industrial application and analysis.

A Description of the BERSAFE System

T.K. HELLEN
Central Electricity Generating Board,
Glos., U.K.

SUMMARY

BERSAFE is an established computer system based on the finite element method for the stress analysis of complicated structures. The system has been developed within the UK power industry (the CEGB) over a period of 14 years, and because of its generality of scope and continuous development, is regularly used in many different engineering industries in the UK and abroad. The present scope of the system is described together with its availability on different computers in different centres.

1. INTRODUCTION

BERSAFE is a computer system designed to perform the thermal and stress analysis of general structures subjected to arbitrary forms of loading. It has been developed for the Central Electricity Generating Board at Berkeley Nuclear Laboratories in England and has been made available to users outside the industry on either an inhouse usage basis,or under licence on their own computer installations.

The analysis of the wide variety of components which exist in generating plant under diverse operating conditions requires a technique as general and as versatile as the finite element method. Components range in shape from simple plates to complicated solids, shell surfaces and framed structures. Consequently, in order to cover as many situations as possible in one facility, BERSAFE was created as a general system, based on the finite element technique, with a large element library covering two and three dimensional structures, plates, beams and shells. Several types of analysis are available, each using the range of available element types and a large number of other features. Because of the generality of scope required to deal with the variety of power plant components in existence, the system has long been suitable and utilized in many application areas outside the power industry.

The nature of the finite element method requires potentially large amounts of data to be read and produced by the main analysis programs, which can be very time consuming to the user, particularly for three-dimensional and shell analysis. Consequently, considerable effort has been devoted during the system development to automatic mesh generation programs, whereby groups of elements can be generated by specifying only minimal data. Checks on the quality of the data generated can be made by automatic computer plots, viewing structures in whole or in part, and from any point in space.

Corresponding to the large amounts of input data required, many pages of output can be produced. Hence, many selective output suppression options exist. Also computer plots of results in the form of deformation, graphs, or contour plots are obtainable. For three-dimensional structures, suitable sections can be defined (not necessarily flat) within which contours are plotted.

Alternative input/output support is available using visual display units. Interactive mesh generation for two-dimensional structures exists in the POINTA program. Two and three-dimensional meshes may be viewed at the screen, with three-dimensional mesh editing and interactive load and constraint data generation. The display of results in a similar interactive scheme is currently under development. Another, newer mesh generation and modification program is BERMAGIC, which can be used in interactive or batch mode. This module uses a high-level command language to enable the user to define a wide range of operations.

The overall system is supported by a large amount of documentation covering the basic theories, program descriptions, and examples of use.

2. SCOPE OF THE SYSTEM

2.1 General

The overall system consists of many computer modules, each of which performs a specific function. Three main categories exist: data generation, analysis, and data presentation. A flow chart of the principal modules is given in figure 1. The different modules operate on part of, or all, the main element library.

The modules of the system are all controlled individually by their own input data and are linked together by comprehensively-designed data files. The user is free to choose which modules to use for a given problem, and for new or complicated applications it is possible to modify existing

or create new modules in his local environment. Within each module, several parts may exist, each part being a single executable program step on the computer performing a specific function within the module. This enhances the efficiency of usage of computer core and datafiles, for instance the main solution of equations in the elastic analysis is in a separate part to the digestion and checking of input data, the datafile controlling the solution being in the format most suited to the solution algorithm. The executive control of the various modules and parts is efficiently handled by high-level catalogued procedures on the computer operating system.

The main analysis modules are FLHE, for thermal analysis, BERSAFE itself for elastic (Phase II) or non-linear (Phase III) analysis and BERDYNE for linear dynamics. These modules are the most important because the majority of development and scientific expertise lies in them, but the importance of good data generation and results presentation modules has long been realised as many suitable programs have been developed over the years.

In order to keep abreast of the latest technological advances in structural analysis and the requirements of the user community, the modules are all in a continuous state of development, defined by phases and levels. A maintenance system exists whereby production versions can be updated to correct program errors or to enable small modifications to be made, more major changes being reserved for a new level release.

2.2 Types of Element Available

In view of the wide range of structures considered for analysis, a corresponding variety of element types is required. In the main, it has been the policy to use well tried and tested elements, although certain elements have been researched and developed within the system. The main demand has been for two and three-dimensional elements, and so the isoparametric families have dominated, BERSAFE being the first system to use and extensively test out these elements. However, certain shell and beam elements have also been incorporated, mainly in the elastic version. Element types are recognised in the system by a name of up to 6 characters. The initial letter describes the class of elements whilst the following numbers indicate the number of degrees of freedom.

The main plane stress elements (which are also used for mathematical and engineering plane strain) are designated EP4, EP6, EP6F, EP8, EP12, EP12F, EP16, EP18, EP20, and EP24, being isoparametric. EP6 is the constant stress triangle with degrees of freedom in cartesian directions. EP4 is a line element used for reinforcing in, typically, concrete. EP8 is a quadrilateral with linear displacement functions, and EP12 and EP16 are respectively the triangle and quadrilateral with

midside nodes, having quadratic displacement functions. EP16 uses shape functions which are second order incomplete polynomials (serendipity) and EP18 is the complete polynomial (Lagrangian) form. EP20 is the cubic triangle and EP24 the cubic quadrilateral.

The corresponding axisymmetric elements are designated EX2, EX4, EX6, EX8, EX12, EX12F, EX16, EX18, EX20 and EX24, with the axis of symmetry being the cartesian y axis.

For linear elastic fracture mechanics, special elements may be used about the crack tip within which the radial displacement from the tip at any point includes a square root dependence on the distance between the point and the tip, consistent with the local crack tip equations of classical elastic fracture mechanics. These elements are designated EP6F, EP12F, EX6F and EX12F and, apart from the changed shape function, resemble their counterparts with no appended F.

The three-dimensional elements are designated EZ6, EZ12, EZ15, EZ18, EZ24, EZ30, EZ39, EZ42, EZ45, EZ45F, EZ54, EZ60, EZ81 and EZ96 again all being isoparametric. EZ6 is a line element for reinforcing or elastic constraints. EZ12 is a simple tetrahedron with constant stress. The elements EZ24, EZ60 and EZ96 are brick-shaped elements having respectively two, three and four nodes along each side. EZ81 is the Lagrangian form of EZ60. The element EZ45 is a triangular prism, and EZ54 is similar with midface nodes in the quadrilateral faces using Lagrangian polynomials. The elements EZ15, EZ18, EZ30, EZ39 and EZ42 are new elements intended to assist in local mesh refinements.

A three-dimensional special crack tip element, EZ45F, exists in the same manner as in two-dimensions. Variations of the two-dimensional axisymmetric isoparametric elements exist for the treatment of Fourier harmonic circumferential variations in displacement and stress, and also for pure torsion. Plane torsion elements are also available.

Several beam elements exist. A variety of different cross-sectional properties, such as annuli, hollow rectangles, offset centres of gravity etc can be specified for the elements CSB12, GBE12 and PEB12. Each has 2 nodes with degrees of freedom $u,v,w,\theta_x,\theta_y,\theta_z$ per node (global coordinates). Either Euler or Timoshenko beam theory is allowed, and the PEB12 element has a constant curvature. A new element, GBE17, has 3 nodes with the above degrees of freedom at the vertices, and semiloof degrees of freedom at the midside position. Some degree of compatibility with semiloof exists for combined usage. HBE12 is a curved beam element suitable for structures like coils. The nodes may be spaced out along the coil without detriment to accuracy.

Most plate-type applications are adequately covered by using shell elements, but one purpose-built plate element, PL9, exists, based on the Zienkiewicz triangle, although at the present time only displacements and rotations are computed.

The available shell elements are FS12, a facet shell with degrees of freedom, u, v and w at each vertex of a triangle and $\partial w/\partial n$ (n being the normal) at the midside nodes. A high ordered shell element, CS54, also exists with 3 nodes and 18 degrees of freedom per node, including zeroth, first and second derivatives of displacement. This particular element is restricted to constant curvature problems. A more versatile type of element is the semiloof element, in triangular (CS24) or quadrilateral (CS32) form. At the vertex and midside nodes, the degrees of freedom are u, v and w, in a global sense, whilst two extra degrees of freedom exist at the midside node representing $\partial w/\partial n$ at two positions (at loof nodes) along that side. These two elements are based on isoparametric formulations, involving shell theories with discrete Kirchhoff hypothesis, and are convenient to use, particularly because the peripheral programs of the system are easily extended when introducing semiloof.

An earlier, closely related, family of thick shell elements of the Ahmad type are also available . They are CS30, CS40, CS50 and CS60 with 5 degrees of freedom per node (u, v, w, in global coordinates and Θ_x, Θ_y in local coordinates).

Numerical integration is used in the evaluation of the element stiffness matrices for most elements in the system. The order of rule required is an input option, in a quantity known as NGAUS. Thus, NGAUS = 3 signifies 3 x 3 or 3 x 3 x 3 in two and three-dimensions, respectively. Complete and reduced integration is therefore easily available, and in certain cases there exist special rules, such as NGAUS=13 or 14 for quadratic solid elements.

2.3 BERSAFE Phase II for Elastic Analysis

Because of user demand, this part of the system is the most highly developed and extensively used. A large number of facilities and all the above element types are available, although each user problem only requires a relatively small number of these options. The most recent level, BERSAFE Phase II level 3, contains many new facilities over the previously issued level 2 in 1976. A description of the main contents of this release is given below.

The input data has been designed so that the user

specifies exactly what he requires and no more. As with other modules in the system, this data is broken into units of data each designed to fit onto an eighty character input card containing a unique card type number. This structure and phraseology are retained, even though nowadays most users have editing screens available. Either fixed format or free format input schemes are available. Default values of certain quantities can be specified to minimise any repetitive data.

Dynamic allocation of storage in the large computer systems now available is used, so that few limits exist in the various structural parameters. However, a limit of 10000 is imposed on node and element numbers for one substructure, due to fixed format input, a limit which can be relaxed using free format schemes.

The solution of equations is effected by the front solution, developed by the author in 1968, and containing versatile housekeeping routines to prepare the element-ordered data for the solution in a previous part of the program. The algorithm stores topology data as node lists against elements and substructures, and yet degrees of freedom may vary from node to node for mixed element types. Any degree of freedom may be decoupled, related to others using generalised constraints, or rotated to operate in individual nodal coordinate systems. Degrees of freedom may be prescribed to zero or some non-zero quantity. The generalised constraints facility allows nodes to be equated to one another, as occurs in sectorial symmetry, or for faces to deform normally whilst remaining flat, or to simply impose the classical theoretical displacement behaviour to degrees of freedom surrounding a crack tip.

A decoupling feature enables any number of degrees of freedom at any node to be independent between different specified groups of elements meeting at that node. This is particularly useful for shear slip problems and for representing cracks of arbitrary shape in structures.

The new substructuring facility is multilevel in nature, with any number of substructures being generated or assembled in a single run. Repetitive stiffness data can be affected trivially if in consecutive topology order, or by calling the same substructure off a datafile when required. Substructures may be mixed with any standard elements, and consist of any number of degrees of freedom on its boundary in the range 1 to the semi-bandwidth limit in the module, which is currently 1500. A new program module is being developed to operate on stored datafiles of substructures, to manipulate them, change their numerical values, rotate them, form them by user-given data, and many more options to enhance the scope of this facility.

For the front solution, the housekeeping routine works to keep the semi-bandwidth as small as possible, allowing this quantity to vary through the solution to maximise computational efficiency. Bandwidth optimisation routines are available in earlier modules. A roundoff criterion is available to check on the deterioration of solution accuracy, which may become dominant in ill-conditioned problems.

Geometry and loading data may be independently specified in cartesian, cylindrical or spherical coordinate systems, irrespective of element type used. Degrees of freedom can also be defined in these systems at the structural or element level, and can be given individual local axes at each node. Stresses are produced optionally in the above systems as element or nodal average stresses, Gauss point stresses, principal stresses with the Von Mises equivalent stress and direction cosines.

A variety of different loadings may be applied to any part of the structure, with up to 50 independent loading cases per run. These include point loads, line loads with facial pressures, and applied strains defined within an element. Centrifugal loading may be applied to the entire structure about axes dependent on the element type, and body loads may be applied in any of the cartesian directions. Nodal temperatures may be specified to induce thermal strains relative to some datum temperature, either from nodal geometry input cards or at any required time from a preceding FLHE temperature transient analysis. Constraints are defined by specifying displacements as zero or some prescribed non-zero quantity, when the corresponding reactions may be calculated. Substructures may be loaded by any of these schemes when generated, or by point loads when being assembled from a file. Independent material properties can be defined for different materials and temperatures, including both isotropic and orthotropic behaviour.

In fracture mechanics, the accurate evaluation of the stress intensity factor, K, at crack tips has been investigated and several techniques for both two and three-dimensional cracks have been developed for Phase II. These include the virtual crack extension (VCE) method, where small crack extensions are specified and, in addition to determining K, the potential energy release rate, G, is obtained plus an indication of which direction the crack will travel under fatigue cases with mixed mode loading. The VCE method uses a special substructuring procedure whch stores stiffness data for the entire structure except crack tip elements. Reruns with further small extensions may therefore be made, at any node along a crack profile in three dimensions, without a total recomputation of the structure.

Displacement substitution methods are based on inferring K by comparing the finite element displacement results at

nodes around the crack tip with the classical crack tip equations. Contour integrals of the J and J* type may be calculated via the ELOPPER module. Several other techniques for evaluating K also exist.

2.4 BERSAFE Phase III for Non-Linear Analysis

This phase performs non-linear analysis using plasticity or creep theory for arbitrary structures in two and three dimensions. Small strain theory is assumed, although a geometry updating facility exists to permit the treatment of large displacements. The input to the non-linear part comprises various indicators, to express requirements of yield criterion (eg Tresca, Von Mises etc), tolerances, time or strain hardening for creep, etc. plus the non-linear materials data. This can be given either as coefficients in a given law of the Ramberg-Osgood type, or as tabulated points, and is sufficiently versatile to deal with nearly all materials of interest. Special laws exist to deal with concrete creep and graphite. Kinematic hardening for plasticity is available through overlay models. The rest of the input data is exactly as in Phase II, including all mesh data, elastic materials, loading, etc.

Results can be optionally stored on a data file, which can subsequently be used for selective outputs and post-processing via the program PLOPPER. Also, this file can be used for restarting. At restart time, a transfer from plasticity to creep or vice versa can be made, and in plasticity a tangent stiffness matrix can be calculated. During the load incrementing process, the initial stress method is used for plasticity and initial strain method for creep, each being based on a re-solve extension of the front solution.

As well as dealing with mechanical and steady state thermal loads, temperature transient effects may be dealt with using a file produced from FLHE. In creep, the time parameter on that file is related directly to the time in the creep process, whilst in plasticity the time parameter is assumed to vary by a user-prescribed amount in each of a sequence of restart runs.

The currently available version of Phase III is level 2, which, as Phase II level 3, uses dynamic storage for the two and three-dimensional isoparametric element families. Stresses and strains are calculated at Gauss points, 2 x 2 or 2 x 2 x 2 for the usual quadratic displacement elements, where they are at their most accurate. This is important since the progress of the solution depends on a continuous assessment of these quantities. Together with nodal displacements, they form the basis of the stored data on the results tape at each load or time point.

A new level 3 version will be available shortly containing many of the new facilities of Phase II level 3 together with further non-linear facilities, including an improved large displacement formulation.

2.5 FLHE

Temperature distributions are calculated by means of the complementary program, FLHE, for either steady state or transient thermal loading problems. The transient problem is solved using a Crank-Nicholson procedure, assuming that the rate of change of structural temperature varies linearly with time between successive computations. The actual heat conduction equations at each reference time point are solved using the same front solution algorithm as in BERSAFE. The user can specify time-dependent loadings as internal heat generation, surface flux, forced convection, free convection, radiation. Temperature dependent flux and internal heat generation can also be specified. Radiation can be to a surrounding medium or between structure surfaces. For the particular case where the gap between surfaces is narrow, heat transfer can also be by forced or free convection, together with conduction across the separating medium. In many cases, because of temperature dependence, the problem is non-linear. The program performs a pseudo linearisation, using the previously calculated temperature distribution, to set up element property and loading matrices for the next computation.

Results from a FLHE analysis can be stored on a datafile as a record of nodal temperatures for each of the given sequence of times. This file may then be accessed in subsequent linear or non-linear stress analysis steps.

2.6 BERDYNE Phase 2

BERDYNE is the complementary version of BERSAFE for linear dynamics analysis. Most of the features and element types available in BERSAFE are also available in BERDYNE, and the use of carefully planned datafiles throughout the overall system ensures access to all pre- and post-processing programs. Phase II level 3 of BERDYNE will be available shortly to correspond to BERSAFE Phase II level 3.

The solution technique is based on nodal condensation, whereby the totality of degrees of freedom which exist in a structure are represented by a relatively smaller number of master degrees of freedom, obtained at the end of the forward elimination stage of the front solution. Current limits on the number of masters is 500, and the remaining slave degrees of freedom are condensed out in a manner which ensures minimum strain energy. Either user-prescribed or automatic choice of masters is available.

BERDYNE consists of 6 parts, each of which is executed in sequence on the computer. Pauses for inspection may be made between any part, all relevant transfer data being stored on datafiles. The first two parts are dedicated to reading input data, checking for errors, combining this with data from pre-processors, and setting up datafiles in suitable form for the subsequent calculation steps, similar to BERSAFE and FLHE. Part 3 goes through each element of the structure, forms stiffness and mass matrices, and performs the front solution for nodal condensation. Part 4 solves the eigenvalue problem to determine the natural frequencies and mode shapes, whilst Part 5 solves the equations of motion. Part 6 deals with outputs and sets up a file for BERSAFE to recover the full displacement pattern.

A variety of different load types can be prescribed, such as harmonic forces, forces expressed as analytic functions or a Fourier series, impulsive forces, forces expressed as sections of cubic polynomials, and support accelerations.

2.7 Treatment of Input

2.7.1 Mesh Generation and Data Modification

The need for automatic mesh generation has been realised since the early days of finite elements, because of the large amount of data involved. Since large areas of mesh are often divided regularly into elements, minimal data requirements specifying suitable boundary nodes and numbers of elements within that boundary are sufficient to give a simple and effective mesh generation scheme. A macro-block is defined as such a region of the structure, which is a convenient unit for automatic mesh generation. Several macro-blocks can then be defined to complete the structure, with micro-data for individual nodal geometry and element topology to fill in any remaining gaps. Alternatively, a coordinate digitising table may be used, for previously drawn meshes, to produce pointwise geometry as card images. Both nodal geometry and element topology can be generated in this way, which is particularly suitable for two-dimensional structures or three-dimensional sections of complicated shape with irregular boundaries, or when macro-blocks cannot be readily defined with regular or graded mesh patterns.

A versatile approach to general two and three-dimensional mesh generation is given by the program BERGEN. The macro-block concept is used, with different types of elements and materials allowed in different blocks, with interspersed micro-data. Adjacent macro-blocks have common node numbering. Each macro-block can be subdivided into mesh patterns, with grading of individual element sizes if required, and the shapes of the blocks can be arbitrary with plane or curved faces. Thus, very complicated shapes can be generated.

A newer approach to the problem of mesh generation and data modification is available using BERMAGIC, compatible with the level 3 main analysis programs and encompassing most of the above processes. The module can be used either interactively or in batch mode on the main frame or local mini computers.

The input data is given in a high level, free format command language which covers the basic operations required to generate and alter meshes. For instance, meshes can be transformed, rotated, doubled up, joined to other meshes, and generated by macro-type instructions. Associated graphics for use at the VDU or hard plotter are available for checking the quality of the generated data. A novel semi-bandwidth reduction algorithm is included in BERMAGIC.

For regularly shaped structures, it is advantageous to have specialised generation modules such as BERCYL, for cylinder-cylinder intersections, of varying relative dimensions. Minimal input data is required, and meshes are produced with refinements judged from experience to give sufficiently accurate results using only moderate numbers of elements.

2.7.2 Interactive Graphics

In recent years, mini-computers typified by the GEC 4080 range have been installed at the principal CEGB research and design centres, with visual display equipment usually of the Tektronix type. A fully interactive data generation system, POINTA, has been developed which offers considerable advantages to mesh design. Meshes of two and three-dimensional elements may be generated and viewed at the screen, or existing meshes modified. Various options exist, such as element and node numbering, window views for investigating selected areas of the mesh, local refinement, topology re-ordering etc. Other types of input for the analysis modules may be generated at the screen, and the final product is a file available for subsequent processing by the main analysis modules.

Developments are in hand to extend the scope of the interactive graphics modules to deal with the presentation of results.

2.8 Treatment of Output

2.8.1 BERPLOT

Owing to the large amounts of output data which can be produced from the main analysis modules, it is important to have a versatile facility to plot the principal results on hard plotters. This enables data checking, results interpretation and often direct use in reports etc. Consequently considerable effort has been extended into the development of

such a program, BERPLOT.

Mesh plots may be obtained for all types of structure, with various options such as selective elements, element shrinking (a useful technique for ensuring that internal elements have not been omitted), node and element numbering, Gauss point locations and numbering for different integrating rules, and views from any point in space onto any prescribed focal plane. In three-dimensional meshes, inside faces and hidden external faces may be optionally omitted. Deformation plots, with degrees of freedom magnified as required, may be superimposed on the original mesh plot, or just the boundary, to give a clear impression of the overall distortion, using, if required, different coloured pens. A much more realistic view of three-dimensional structures, shells, etc. can be obtained by the use of a steroscope, simply effected by taking two plots with viewpoints sufficiently close together to simulate the position of the eyes.

Contour plots are available for plotting stresses, strains, temperatures and displacements as contours of given or automatically calculated magnitude in two-dimensional planes, together with a suitable boundary plot of the structure. For three-dimensional structures, suitable planes, curved surfaces or outside surfaces can be defined for such plots, which can be viewed from any point in space. As usual, selected areas only need be plotted if required, and clarity is enhanced by the use of two-coloured plotting. Stresses and strains are plotted using either nodal values or Gauss point values, which is particularly useful for results produced by the non-linear version BERSAFE Phase III. Yield zone boundaries at different load levels for plasticity may also be plotted.

Mountain plots are available for plotting any component as above in the z direction over a two-dimensional mesh. The resulting plot consists of the original mesh and the deformed mesh, viewed from any point in space.

An alternative form of plotting is scalar plotting, which produces values of a given component such as stress or displacement along a line of nodes or Gauss points through the structure, or temperatures against time. Since the line may traverse freely through the structure, such plots are particularly suitable for using directly in reports.

BERPLOT has been modified to give results directly onto visual display units. A similar program called LINOUT produces scalar plots directly onto printed output, and is useful for a rapid appraisal of results.

2.8.2 ELOPPER

ELOPPER is designed to print results stored on a file produced by BERSAFE Phase II. This enables various printouts of selected results and avoids the necessity of large amounts of output during the BERSAFE calculation. Additional features include the computation and output of strains, together with a facility to store them for subsequent contour plotting in BERPLOT, and facilities for combining stressing cases from different runs and files and creating a new file, of the same format as that produced by BERSAFE Phase II, containing these combinations. J and J* integrals for linear elastic fracture mechanics may also be calculated for two and three-dimensional structures along contours chosen by the user.

2.8.3 PLOPPER

PLOPPER is designed to process results obtained during a plasticity or creep computation using BERSAFE Phase III. Selective output control is again available, thereby reducing the need to produce large amounts of output during the relatively costly non-linear computation step. Also, nodal stresses and strains are derived from the neighbouring Gauss point values used in the non-linear computations. Together with yield zone boundaries, these may then be written onto files for subsequent plotting via BERPLOT. J and J* integrals for non-linear fracture mechanics may be calculated for two and three-dimensional structures along contours chosen by the user.

3. AVAILABILITY OF BERSAFE IN INDUSTRY

The versatility and wide scope offered by the overall system has encouraged its release for outside usage. Over the last 12 years, the system has been available under licence for use at the customer's own installation, with full source programs and documentation. The number of such licencees has increased gradually, most being in organisations in Europe, in Italy, Belgium, Denmark and Yugoslavia, and the UK, although licences also exist in India and Australia.

Within the UK, most users in organisations outside the CEGB prefer to use the hardware facilities of the CEGB, usually via a remote terminal at or near their own centre, thereby removing the need to purchase the system for their own in-house use. A large number of organisations take advantage of this facility, including the manufacturers of engineering components relevant to the power industry. The cost of use only covers computer costs, all technical consultancy being free.

Two other centres in the UK also permit the use of BERSAFE on their hardware installation to external users. The Science and Engineering Research Council has mounted the

system at the Rutherford Laboratories Computing Centre, with remote terminals at many Universities. J. Lucas Ltd have developed a computer aided design package around their version of BERSAFE which they market under the name of FELSET.

The most widely used areas of the system are the elastic stress analysis function in BERSAFE, FLHE and the pre and post processors. Mesh generation programs are heavily used, with BERGEN and BERMAGIC being the most popular. However, the advent of visual display units over the last 2 or 3 years has meant that the POINTA program is now being used more and more, either on Tektronix 4010 or 4014 type units attached to GEC 4080 series mini computers, or by direct links to the main computer installation in London. These mini-computers are installed in several areas of the CEGB, such as the regional Scientific Services Departments, the Research Laboratories, and other departments, and all have communication links with the main computer installation, so once data has been prepared locally, it can be transferred to London immediately for the main analysis modules. Their recent availability has also encouraged other pre and post processing operations to be performed locally rather than on the main installation.

BERSAFE has been used mainly on IBM 370 computers, since these have always been installed at the Computing Centre of the CEGB. The present installation comprises an IBM370/168, an IBM3032, an IBM3081 and an Amdahl 470/V8, all linked together, with virtual storage core and time sharing. The system takes advantage of the large fast core availability (about 8 megabytes on each mainframe) by using dynamic storage of program arrays, and therefore the new levels of the system have very relaxed limits on the size of problem that can be analysed.

Despite this large core availability, the programs have been written with a view to machine independence, and in particular the dynamic storage techniques ensure equally efficient use on computers of limited core size. Thus, mini-computer installation is very straightforward, although increased efficiency would result by removing any parts of the programs which are not required.

Apart from the CEGB installation, BERSAFE has been converted to and used regularly on UNIVAC 1100 series, ICL 1900 series, Burroughs, Prime (a mini-computer), and VAX computer systems.

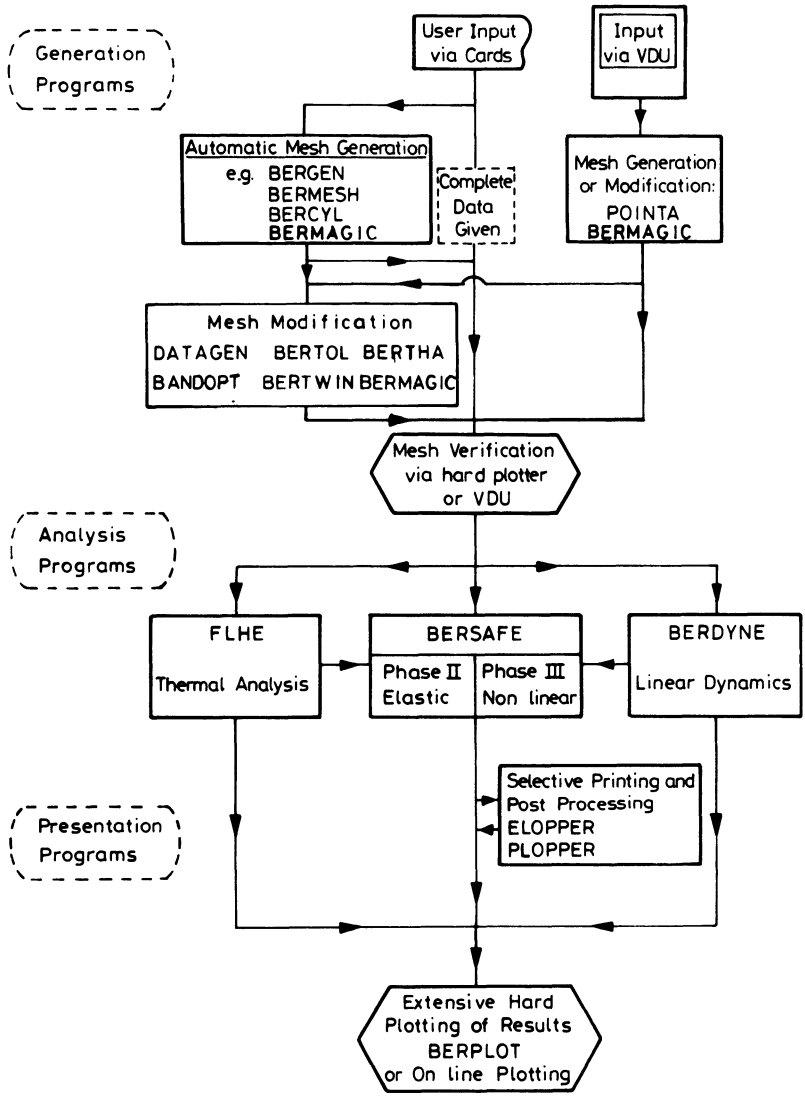


FIG. 1. Flow Chart of the Overall System

FLUID MECHANICS USING FINITE ELEMENTS

P. Ward

S.D.R.C. Engineering Services Limited
Hitchin Hertfordshire SG4 9DY

There has been a significant advance recently in the application of finite element methods in the field of laminar and turbulent incompressible flow. This paper describes the theoretical basis of SDRC's FLODYN program for two dimensional and axi-symmetric flow and describes some of the programs capabilities.

THEORETICAL OVERVIEW

The solution of the Navier-Stokes equations for viscous incompressible flow requires the consideration of two classes of differential equation.

Firstly, the condition of incompressibility states that

$$u_x + v_y = 0 \tag{1}$$

This equation represents a kinematic constraint on the cartesian velocity field (u, v).

The second class of equations are the differential equations specifying the conservation of momentum

$$u_t + uu_x + vu_y = -\frac{1}{\rho} P_x + \nu \nabla^2 u \tag{2}$$

$$v_t + uv_x + vv_y = -\frac{1}{\rho} p_y + \nu \nabla^2 v \tag{3}$$

These equations are nonlinear due to the presence of the convection terms. To obtain a solution we must consider the coupling between these equations. Boundary conditions for u, v and consequently p must be included consistently. In practice, however, it is found that coupling between pressures and velocities introduces difficulty both for the coupled numerical integration of the system and the introduction of

boundary conditions.

A second approach involves the introduction of vorticity as a new variable replacing equations (2) and (3) by the vorticity transport equation.

$$v_x - u_y = \omega \tag{4}$$

$$\omega_t + u\,\omega_x + v\,\omega_y = \nu\,\nabla^2\,\omega \tag{5}$$

Pressure has been eliminated as a variable and an additional kinematic relationship (4) has been introduced leaving just the vorticity transport equation to be integrated. The problem now reduces to one of obtaining a coupled solution of the boundary conditions. Again, difficulties are usually encountered in obtaining full coupling, particularly at the boundaries.

A VARIATIONAL APPROACH

A variational form for this problem was introduced in (1). The kinetic energy of an Eulerian volume of fluid is minimised under a set of constraints using the following Lagrangean.

$$L = \tfrac{1}{2}\,(u^2 + v^2) + \phi\,(u_x + v_y) + \beta\,(-v_x + u_y + \omega)$$

$$+ \eta\,(\omega_t + u\omega_x + v\omega_y - \nu\,\nabla^2\,\omega) \tag{6}$$

After the Clebsch transformation of the form

$$u = \phi_x + \beta_y - \eta\omega_x \tag{7}$$

$$v = \phi_y - \beta_x - \eta\omega_y \tag{8}$$

a new Lagrangean may be written as follows

$$L^* = \tfrac{1}{2}\,(\phi_x + \beta_y - \eta\omega_x)^2 + \tfrac{1}{2}\,(\phi_y - \beta_x - \eta\omega_y)^2$$

$$- \beta\omega + \nu\eta\nabla^2\omega - \eta\omega_t \tag{9}$$

The advantage of this approach lies in the boundary conditions (2). The boundary conditions for ϕ and β are simply the natural boundary conditions for normal and tangential velocity

$$\delta\phi\text{: } \int (\underline{u} \cdot \underline{n} - f)\,\delta\phi\,ds = 0 \tag{10}$$

$$\delta\beta\text{: } \int (\underline{u} \cdot \underline{t} - g)\,\delta\beta\,ds = o \tag{11}$$

where f and g are specified velocity distributions on the boundary s. This formulation thus provides the flexibility of defining boundary velocities as natural boundary conditions, a technique which has been used widely for potential flows.

Variation with respect to η does not yield a boundary term while variation with respect to ω yields a boundary condition which is coupled between ω and η

$$\int \eta^2 \left[u_n \left(\frac{\delta\omega}{\eta}\right) - \nu \frac{\partial}{\partial n} \left(\frac{\delta\omega}{\eta}\right)\right] ds = o \qquad (12)$$

This boundary condition implies that for abitrary values of vorticity ω, the lagrange multiplier η must satisfy a constraint on the boundary. In the case of solid walls, the vorticity transport equation becomes unnecessary and the vorticity at the wall can be determined from the kinematic relationship (4). This may be achieved by setting the Lagrange multiplier η to zero at the walls. In the case of an inlet or outlet with a specified velocity distribution (12) provides the natural boundary condition as the flow converges to a steady state.

THE FINITE ELEMENT TECHNIQUE

The functional of equation (9) is discretised using finite elements in space and time. The nodal variables are the Lagrange multipliers ϕ, β and η together with the Vorticity ω. Minimisation with respect to the nodal variables produces a set of algebraic equations to be solved by marching in time.

Modelling Techniques

Modelling a flow domain using finite elements has similar pre-requisites to the well established applications of the finite element method in other fields. A modeller uses his knowledge of the physics of the problem under consideration to produce a suitable mesh. It is essential that the mesh be biased towards regions in which the variables vary most rapidly. For viscous flow fine resolution close to solid boundaries will be required to resolve the steep gradients present in the boundary layer. The mesh can be much coarser outside the boundary layer as the spatial gradients of the solution variables are generally smaller. When modelling flow involving sharp changes in geometry, resolution is also required to represent steep gradients of variables in the flow direction.

The finite element method is particularly suited to the representation of complicated geometrical configurations and the availability of well developed model preparation software such as SDRC's SUPERTAB program greatly aids the effective use of the technique

SDRC's FLODYN PROGRAM

The theoretical approach outlined in this paper forms the basis of SDRC's FLODYN computer program. This program represents a radical departure in the field of finite element software. The system appears to the user as a menu driven interactive package built around a version of SDRC's well established finite element pre and post processing software SUPERTAB and OUTPUT DISPLAY. The interactive software is used to generate the finite element mesh and fluid mechanics boundary conditions, specify salient problem parameters such as solution method, time step or relaxation factor and initiate an analysis by issuing a command to SOLVE. This command automatically calls the finite element program, which forms the kernel of the system, for interactive or deferred batch execution.

Solutions are obtained by a process of numerical integration corresponding to marching in time in a sequence of time steps. The user specifies the number of time steps to be performed and on completion the program automatically stores the current solution in a form appropriate for display. It is not necessary to submit an analysis for the total required number of time steps as restarts are fully automated, requiring no user intervention regarding the assignment of files, etc. In fact, the user is isolated from file assignments and data handling which forms a part of more traditional finite element systems.

Finite Element Solver

The finite element solver, which is at the heart of FLODYN, is highly modular and its execution sequence is controlled from a command file, with its own execution control language.

The primary objectives which motivate the development of the execution control language were as follows:

(a) To provide an extensive menu of analytical functions to enable experienced analysts to design custom-built procedures which address the unique requirements of specific applications. This was considered mandatory in such a technically sophisticated application as it was considered unnecessarily restrictive to attempt to anticipate all the potential uses of the software.

(b) To provide a means of accessing and executing standard solution procedures while still maintaining a highly modular software design whereby algorithm structure is totally isolated from the software architecture.

The execution command language supports those features required for conditional branching and construction of DO LOOPS within the control system. An example command file is illustrated in Fig 1

Several standard solution procedures are available

- Potential flow
- Solution for streamlines
- Solution for the pressure field
- Solution for a globally smoothed velocity field
- Two alternative solution algorithms for viscous incompressible flow

The results from these analyses are automatically archived in the data base for graphical display and, where appropriate, restarts.

Turbulent Flow

FLODYN is also capable of obtaining solutions of the time averaged Navier-Stokes equations for turbulent incompressible flow. This requires the specification of a turbulence model. Presently the program supports two mixing length models one for attached boundary layers and the other for wake regions. Turbulence is then represented using the concept of an eddy viscosity. Further details are contained in (2)

TYPICAL RESULTS

Several internal and external flow examples have been analysed using FLODYN. The program supports both planar two dimensional and axi-symmetric flow cases.

Flow in a channel with a backward facing step has been analysed over a range of Reynolds Numbers using the finite element grid shown in Fig 2. Fourteen elements were used in the flow direction with eight elements and fourteen elements across the duct upstream and downstream of the step respectively. The flow Reynolds Number is based on the step height and the average inlet velocity. Results are presented for laminar flow (Re=73) and turbulent flow (Re=3025) and compared with others (3-5).

As mentioned previously, FLODYN solves the unsteady Navier-Stokes-equations for mean flow quantities. The time dependent behaviour of vorticity at user specified grid points is automatically monitored for graphical display in order that a user has access to a visual check of the convergence history of his problem. This information quickly highlights the choice of an inappropriate time step. The unsteady nature of the flow and the stability of the present numerical scheme has been discussed peviously (6).

REFERENCES

(1) Ecer, A. Variational Formulation of Viscous Flows, Int. J. for Numerical Methods in Engineering, Vol 15, 1980, pp 1355-1361

(2) Ecer, A., Rout, R.K. and Ward, P., Investigation of Solution of Navier-Stokes Equations using a variational formulation to be published in Int J. for Numerical Methods in Fluids January 1983

(3) Taylor, C., Thomas, C.E. and Morgan, K. Confined Turbulent flow utilising the Finite Element Method, Finite Element Methods for Convection Dominated Flows, Ed. T.J.R. Hughes, AMD Vol 34, ASME, 213-224, 1979

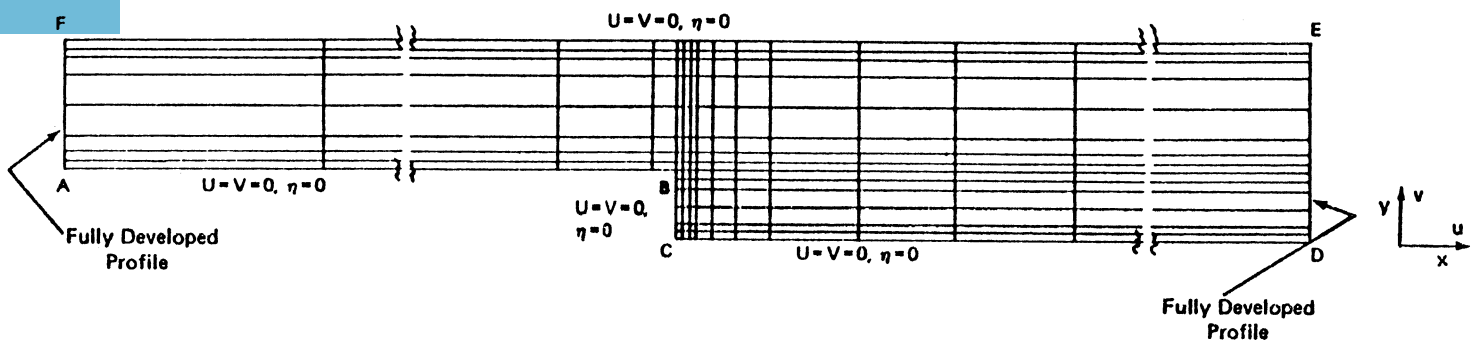
(4) Atkins, D.J., Maskel, S.J. and Patrick, M.A. Numerical Predictions of Separated Flows, Int. J. for Numerical Methods in Engineering, Vol 15, 129-144, 1980

(5) Denham, M.K. and Patrick, M.A. Laminar Flow over a Downstream facing step in a two dimensional Flow Channel, Trans. Inst. Chem. Engrs. 52(4), 361-367, 1974

(6) Ecer, A., Ward, P. and Akay, H.U. Finite Element Analysis of Convective and Viscous Flows, Finite Element Flow Analysis, ed T. Kawai. Proceedings of the Fourth International Symposium on Finite Element Methods in Flow Problems held at Chuo University, Tokyo, July 26-29, 1982

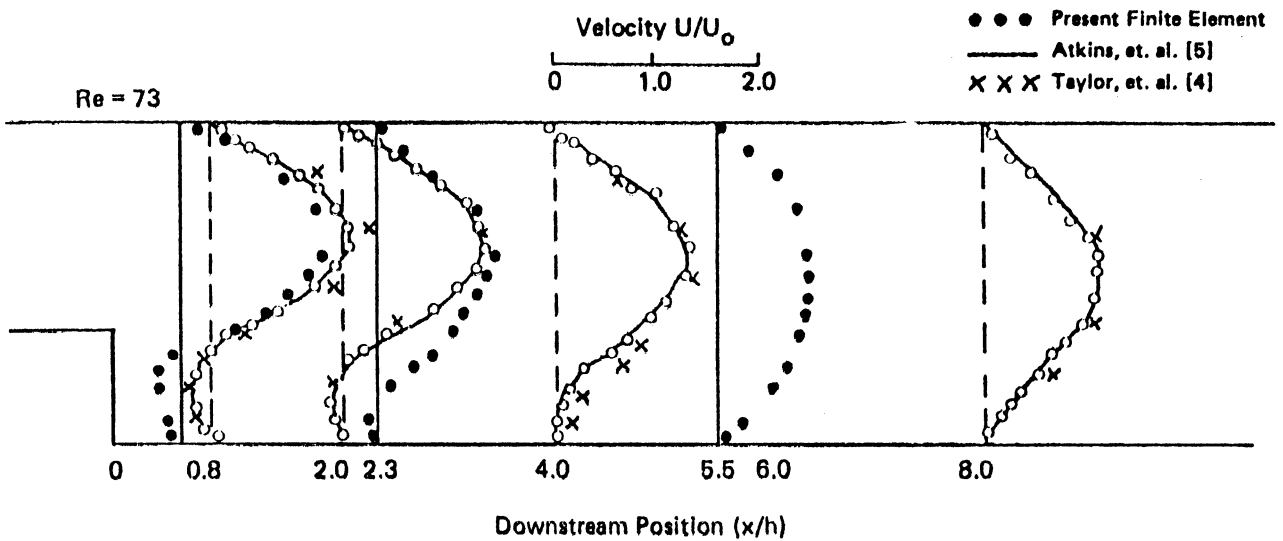
```
*C
*C     UPDATE MATRIX METHOD
*C
*C    FLODYN VERSION 1.0
*C
*C   I41=NUMBER OF STEPS TO RUN
*C   I42=FLOW TYPE (=0 LAMINAR)
*C
*ASGN  I1 = I41
SET,PRINT_LEVEL,4
FORM,TABLE
FORM,BOOLEAN
PIVOT,3,4
RESTORE
FORM,ELEMENT_VECTORS,DEFINED
*LAB 10
ITERATION,START
*JEQZ I42 5
FORM,TURBULENCE_DATA
*LAB 5
FORM,ELEMENT_MATRICES
FORM,ELEMENT_VECTORS,LOADS,EXPLICIT
ASSEMBLE,GLOBAL_MATRIX
ASSEMBLE,GLOBAL_VECTOR
DECOMPOSE,CROUT
SOLVE,NONSYMMETRIC
FORM,ELEMENT_VECTORS,CURRENT_SOLUTION
FORM,ELEMENT_VECTORS,CENTROIDAL_VELOCITY
ITERATION,END
*ASGN I1 = I1 - 1
*JGTZ I1 10
SAVE,CENTROIDAL_VELOCITY
PRINT,SOLUTION
PRINT,MESSAGE,'***JOB COMPLETE***'
CLOSE
```

Fig 1. Execution Command File



Finite Element Model for Step Duct

Fig 2.



Axial Velocity Profile for Step Duct (Laminar Flow)

Fig 3.

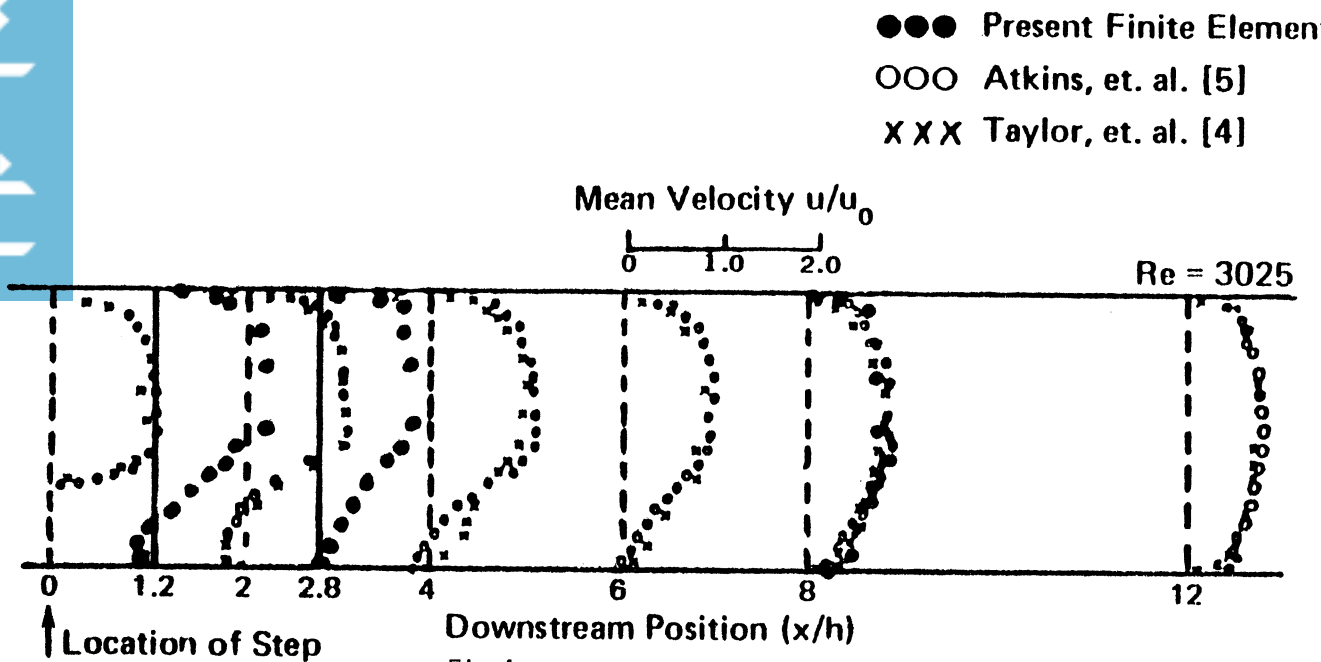


Fig 4.

FINITE ELEMENT UPSETTING ANALYSIS OF A RING: AN INCREMENTAL SOLUTION TO THE CONTACT PROBLEM

Ibrahim M. Al-Khattat
Department of Mechanical Engineering
Yarmouk University, Irbid, Jordan

ABSTRACT

The ring test has been devised to measure friction paratmeters under extreme pressure conditions as in metal forming processes. However, gravely erroneous assumptions are used to develop and interpret the test. The continuum difinition of stress is violated not only by the test theory but also by the current concepts of friction and lubrication. The author has proposed a general solution for the contact problem based primarily on the continuum definition of stress as a local quantity. An experimental procedure, following Bridgman's shear-compression fracture tests is proposed to measure frictional properties in a new format. The resulting information is introduced in the analysis as explicit displacement or traction boundary conditions. The incremental nature of the nonlinear boundary value problem involving the deforming media in contact is conveniently used to generate a discretized approximation to the state of stress and deformation along the contact surface at the end of each solution increment. An important consequence of this approach is that the symmetry of the stiffness matrix in the finite element analysis is insured, resulting in considerable computational savings. Another advantage of the proposed method is the elimination of non-physical corner stress singularities. Moreover, recognition of the local nature of contact stresses implies that the proposed method is applicable to any geometry of contact.

The finite element analysis of the upsetting of the standard steel ring is carried out using an elastic-plastic finite deformation theory and the new method of contact modeling. Computer graphics are employed to display stress distributions. The results point out the serious shortcomings of the standard ring test and the present friction measurement methods in general. The proposed method may be used in computer-aided design of metal forming processes and in the analysis of many complex boundary value problems involving contact.

INTRODUCTION

A Ring Compression Test to measure the coefficient of friction μ under conditions of bulk plastic deformation was suggested by Male and Cockcroft (1964). The test method was later extended by Male and Depierre (1970) to measure the friction shear factor m. For both tests, assumptions of no barrelling, no material work-hardening, and the existence of a constant μ or m were the basis of the analyses developed. A uniform friction shear stress directed radially inwards was assumed all along the billet-platen interface. Despite the shortcomings and limitations in both the theories used and the interpretations of the results, as was pointed out by their developers and others, these tests have been widely used as a tool for friction measurement and modeling.

In this paper, the contact condition in the problem of ring compression is treated from a point of view based on the notion of stress at a point in the theory of continuum mechanics. In such a context it is unsatisfactory to assess a stress by averaging a force over a surface even though this might be consistent with the purposes and needs of rigid body mechanics. All present methods of measurement of frictional properties can either be linked to a framework of rigid-body mechanics, such as the Coulomb law, or are the results of unbased or oversimplified assumptions, like the friction shear factor m in plastic deformation.

During an era when analytical solutions of science and engineering problems were the final objectives of analysis, it was essential to make simplifying assumptions and devise parameters that made a solution possible. In the present age of automatic computation, it is interesting and important to identify cases in which those very simplifying assumptions and parameters become a source for complications in the analysis. Friction is one case in question. Current difficulties linked to friction are not confined to the unreliability and inadequacy of the existing models but extend to the computational implementation of some of these models. For example, the use of the Coulomb law leads to a nonsymmetric stiffness matrix in the numerical analysis of friction problems. Moreover, the Coulomb friction boundary condition,

$$\tau = \mu p, \tag{1}$$

relating the frictional shear τ to the normal pressure p, is not covered by the standard proofs of existence and uniqueness of solutions for boundary value problems (BVP's) of deformation, see Goodman (1980).

The capability of computation provides the possibility of using a more realistic representation of the physics of a problem rather than 'simplifying' it through assumed mathematical models. The author (1981), (1982) has proposed a method to measure, model, and numerically implement contact interface

phenomena, i.e., friction, lubrication, and adhesion, collectively termed 'generalized friction'. Tests by Bridgman (1952) on fracture of metals under a combination of compression and twist, when performed on a ring geometry, led to near-uniform distributions of longitudinal and shear stresses. Hence, the AVERAGE stresses measured were very nearly the same as the actual LOCAL stresses. From tests of this type it is possible to find relations between the local frictional sliding shear stress (LFSSS) and the local normal pressure p, under specified conditions of the other friction-related variables, like temperature, sliding speed, and surface texture. Figure 1 shows the type of results that might be produced by such proposed tests.

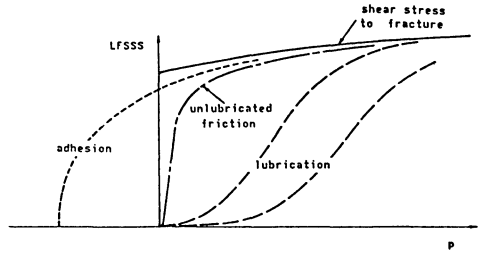


Figure 1 Graphical Representation of Generalized Friction

The local nature of the measured stresses means that they are geometry-independent, so that the test results are applicable to any problem relevant to the particular tests performed. The curves shown in Figure 1 would be generalized to surfaces or hypersurfaces if more variables were included.

The various frictional phenomena, represented by these curves, can be introduced into the mathematical and numerical models of the BVP through a simple algorithm that preserves symmetry of the stiffness matrix in the finite element configuration.

'Perfect roughness' of contact provides a special case of the author's method of dealing with generalized friction. The problem of ring compression between perfectly rough platens is analyzed in order to evaluate the Ring Compression Test itself and to provide a reference with which experimental results and previous solutions using unsatisfactory boundary conditions can be compared.

THE RING UPSETTING PROBLEM

A ring with an outside diameter:inside diameter:height ratio of 6:3:2 (as in the Ring Test) is compressed axially to 50% of its original height, between two perfectly rough, rigid, and parallel platens overlapping it. The problem parameters are listed in Table 1. The material properties are typical of spherodized 0.45% carbon steel.

Table 1
Parameters of the Ring Upsetting Problem

Young's modulus, E	200×10^3 MPa
Initial yield stress in tension, Y_o	700 MPa
Tangent modulus, h (constant)	300 MPa
Poisson's ratio, ν	0.3
Outside diameter	19.050 mm
Inside diameter	9.525 mm
Height	6.350 mm

THE ELASTIC-PLASTIC FINITE DEFORMATION THEORY

The basis of the continuum modeling is the elastic-plastic Prandtl-Reuss rate equations as generalized by McMeeking and Rice (1975) to account for finite deformation:

$$\overset{\circ}{\tau}_{ij} = C_{ijkl} D_{kl} \tag{2}$$

where $\overset{\circ}{\tau}_{ij}$ is the spin-invariant Jaumann rate of Kirchhoff stress, D_{kl} is the rate of deformation, and C_{ijkl} is the constitutive tensor given by :

$$C_{ijkl} = \frac{E}{1+\nu}\left(\delta_{ik}\delta_{jl} + \frac{\nu}{1-2\nu}\,\delta_{ij}\delta_{kl} - \Theta\,\frac{3s_{ij}s_{kl}}{2\bar{\sigma}^2\{1+\frac{2h(1+\nu)}{3E}\}}\right) \tag{3}$$

In Equation (3), δ_{ij} is the Kronecker delta, Θ is unity for loading at yield and zero for eleastic loading or any unloading, s_{ij} is the deviatoric Cauchy stress, and $\bar{\sigma}$ is the generalized stress:

$$\bar{\sigma} = \left(\frac{3}{2}s_{ij}s_{ij}\right)^{\frac{1}{2}} \tag{4}$$

The variational formulation of McMeeking and Rice (1975) is also used in the form:

$$\int_V \{\overset{\circ}{\tau}_{ij}\delta D_{ij} - \tfrac{1}{2}\sigma_{ij}\delta(2D_{ik}D_{kj} - v_{k,i}v_{k,j})\}\,dV = \int_V \dot{b}_i \delta v_i\,dV + \int_S \dot{f}_i \delta v_i\,dS \tag{5}$$

Here, σ_{ij} is the Cauchy stress, $v_{i,j}$ is the velocity gradient, $\dot{b}_i$ is the rate of body forces per unit volume, and $\dot{f}_i$ is the rate of surface forces per unit area, with integration performed over current volume V and current surface area S. This leads to the discretized incremental finite element stiffness equations:

$$(K)\{\Delta u\} = \{\Delta P\} \tag{6}$$

where K is the elastic-plastic stiffness matrix, Δu is the vector of nodal displacement increments, and ΔP is the vector

of nodal force increments.

A four-noded isoparametric quadrilateral ring element with one integration point is adopted. By symmetry, only the upper half of the ring is considered. Figure 2(a) displays the 20 x 11 element mesh.

THE INTERFACE BOUNDARY CONDITION

Perfect roughness of contact is defined to mean that the LFSSS assumes its natural upper bound, i.e., the shear stress to fracture (SSF) of the weaker of the two materials in contact. Bridgman (1952) showed that the SSF for mild steel hardly changes with pressure, so a constant value of the LFSSS is assumed. This corresponds to perfect roughness being represented by the line LFSSS = a constant in Figure 1.

At the beginning of the initial (elastic) displacement increment, a zero displacement increment along the interface is specified at all contact nodes, i.e.,

$$\Delta u = 0 \tag{7}$$

The solution at the end of this increment renders (τ,p) values that define a set of coordinates for a point. If this point falls below the friction curve, the boundary condition of Equation (7) is maintained in the next increment. However, if the point is on or above the curve, a switch to a zero traction increment condition is made at the respective node, i.e.,

$$\Delta\tau = \Delta(\text{LFSSS}) = 0 \tag{8}$$

This means that the node is freed to slip, and Equation (8) therefore represents a first estimate of the LFSSS increment for the following solution increment. It also means that at the end of each increment, complete information about stress and deformation along the contact surface is obtained in a discretized form.

For a non-constant friction curve, iteration would be necessary to obtain an improved estimate of $\Delta\tau$ so that the friction curve is closely followed. Another factor requiring iteration is the change in surface area due slipping under load. In the present problem, however, a correction in $\Delta\tau$ due to change in area is made at the end of each increment for use in the following one, and no further iteration is attempted.

Equations (7) and (8) indicate that the tangential and normal degrees of freedom, tied together in Equation (1), have been decoupled, resulting in a symmetric stiffness matrix.

RESULTS AND DISCUSSION

Surface II computer graphics, see Sampson (1978), are used to display the distributions of stresses and generalized plastic strain over the analyzed section. Both a contour plot and a perspective diagram are presented. Although it is evident that

valuable information can be gained from such drawings, they should only be considered as qualitative representations, because of the nature of the built-in features of the graphics software. All stresses shown are in megapascals (MPa). The azimuth value indicated in the figures defines an observer's viewing angle of the perspective diagram expressed in degrees of anticlockwise rotation from the 'south' of the contour plot.

Figures 2(b) and 2(c) show the deformed mesh at 25% and 50% compression respectively. They indicate significantly greater radial stretching in the elements within the neutral radius (i.e., the radius at which there is zero frictional shear stress) as compared with outer elements.

An interesting feature of the deformation pattern in the early stages is double bulging on the inside cylindrical surface. Traces of this early excessive deformation near the inside billet-platen corner are still in evidence at 25% compression, Figure 2(b). This phenomenon was also reported by Nagamatsu et al (1972).

A simplifying aspect of the present contact model is that the value of the neutral radius is not needed in the analysis and is in fact determined by it at every step of the incremental solution. A neutral radius at r = 6 mm seems to be quite stationary at this value throughout the deformation process.

Figure 3 depicts the axial stress distributions at 25% and 50% respectively. Stress concentrations exist at the billet-platen corners as expected. By the nature of the contact model used, no stress singularities can occur because frictional shear is forced within its upper bound and is relieved by slipping. This manner of modeling is closer to physical reality than assuming no slip to represent perfect roughness, as the latter assumption produces mathematical and numerical models involving non-physical stress singularities at the corners, see Al-Khattat (1982).

The shear stress distributions of Figure 4 show that most of the contact surface is slipping outwards (especially at 50% compression) although local elemental deformation is more excessive within the neutral radius. To assume that the frictional shear stress is uniform all over the contact surface and is directed inwards, as was the case in the Ring Test, is clearly wrong.

Figure 5 shows the total generalized plastic strain at 25% and 50% compression respectively, indicating the most excessive plastic yielding to be near corners and in the centre of the ring section. In the vicinity of the interface by the neutral radius a state of 'hydrostatic' compression prevails within the material instead.

Figure 6 indicates the total upsetting force required against percentage deformation. Also plotted for comparison is the upsetting force for the hypothetical case of a frictionless

interface in which the problem degenerates into a simple one that has an analytical solution, see Al-Khattat (1981).

CONCLUSIONS

Representation of frictional shear stress as a constant fraction μ of normal pressure - an unbounded quantity - can lead to serious errors in the analysis of extreme pressure problems due to the lack of an upper bound on surface frictional shear. The present work further shows that the determination of μ (and the friction shear factor m) from the Ring Compression Test is unreliable because of the invalidity of most of the basic assumptions of the test.

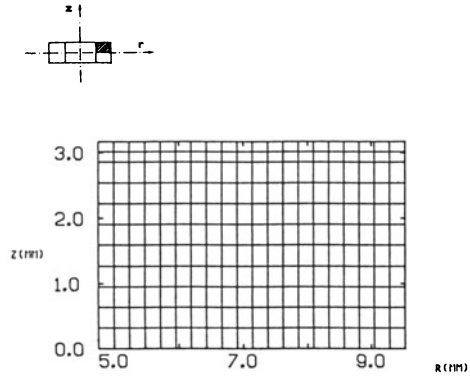
The problem of upsetting of a ring, with the same relative geometry as that in the Ring Test, is analyzed using a perfectly rough billet-platen interface. This contact condition corresponds to a special case of a proposed continuum model for solid interface phenomena.

The proposed interface model offers considerable advantages over current methods of measurement, representation, and numerical implementation of the various interface phenomena.

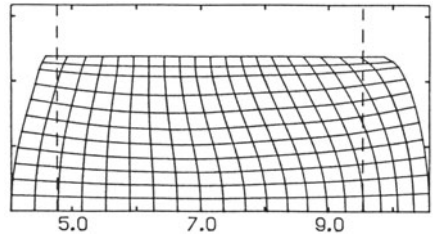
Complete information about the state of interface stress and deformation is obtained numerically at every incremental step of the solution, so there is no need for difficult and costly experimentation (whose results are geometry-dependent anyway), see Nagamatsu et al (1970) and Lu et al (1982) for example. The obtained interface information would be valuable in predicting unfavourable surface conditions throughout a process of deformation, as part of computer-aided analysis and design.

ACKNOWLEDGEMENT

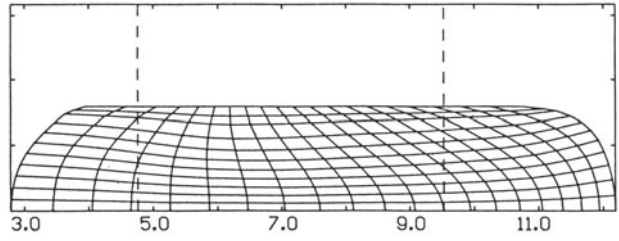
This work was sponsored by the Solid Mechanics Program of NSF under Grant No. NSF-ENG-77-16342 to Stanford University, USA; the author wishes to express his appreciation for this support. The author is also grateful for the help and constructive comments to Prof. Erastus H. Lee and Dr. Russell L. Mallett, and for the use of the computer program to Dr. Robert M. McMeeking.



(a) - Finite Element Mesh



(b) - Deformed Mesh at 25% Compression



(c) - Deformed Mesh at 50% Compression

Figure 2 Finite Element Mesh Configurations

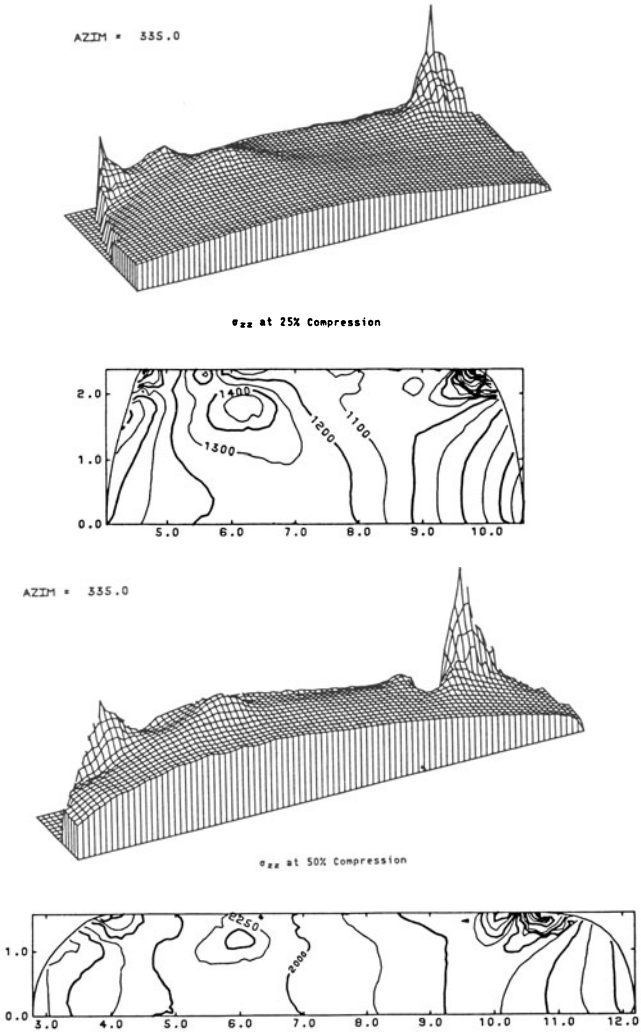


Figure 3 Axial Stress, σ_{zz}

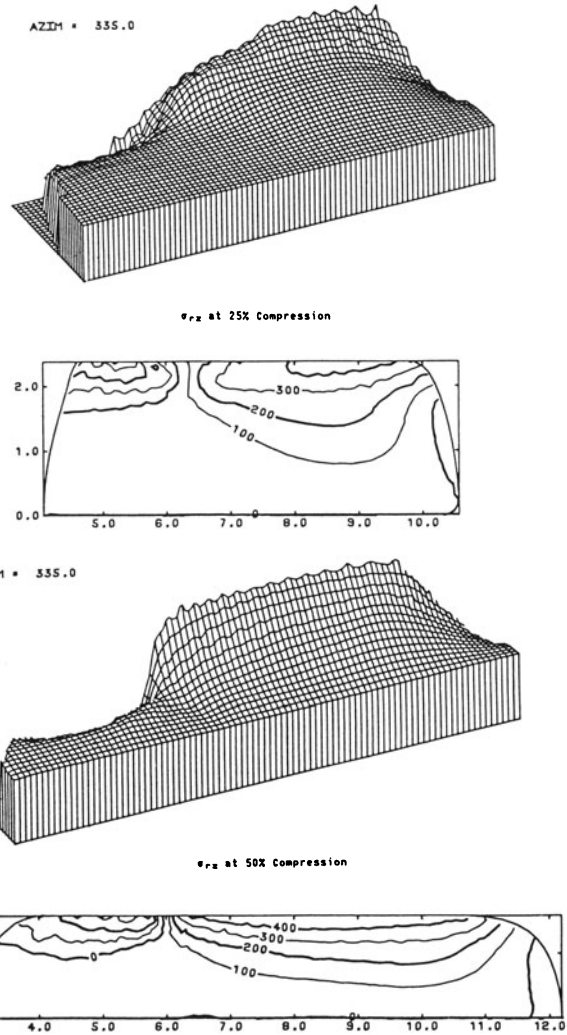


Figure 4 Shear Stress, σ_{rz}

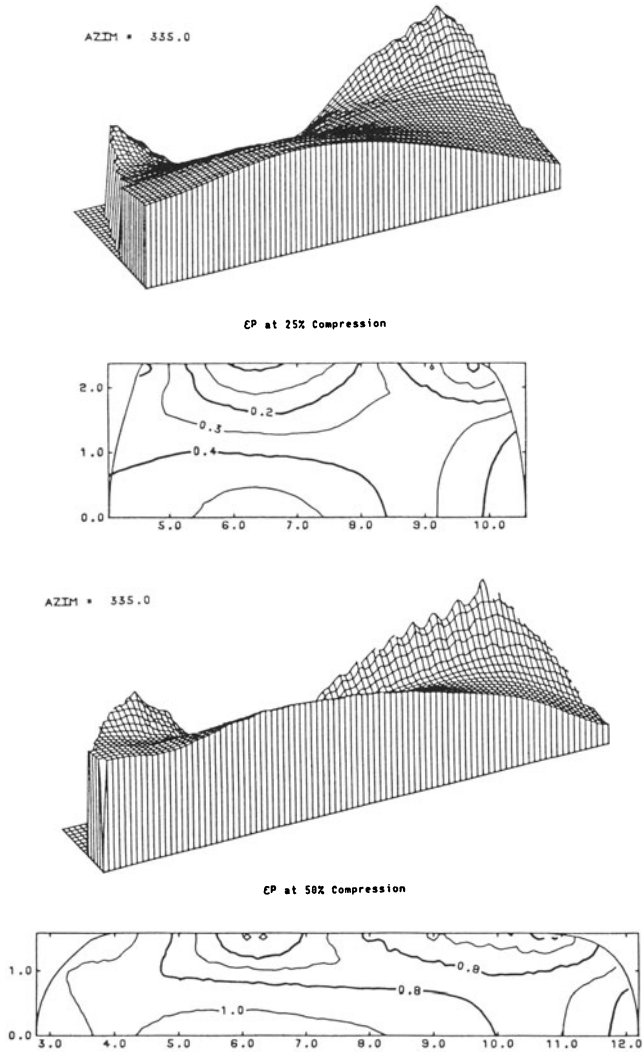


Figure 5 Generalized Plastic Strain, ε^p

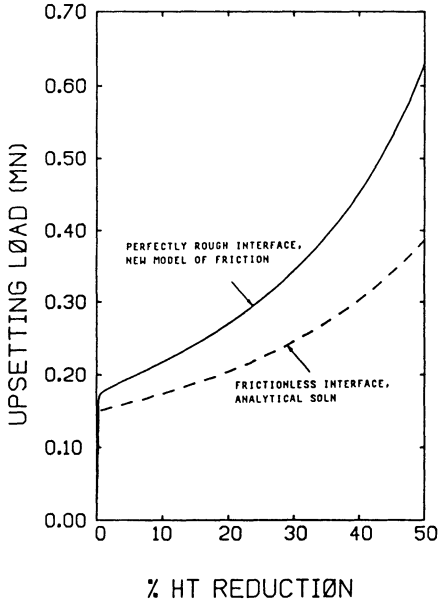


Figure 6 Total Upsetting Force

REFERENCES

Al-Khattat, I.M. (1981) A Continuum Model for Friction with Applications in Metal Forming Plasticity. Ph.D. Dissertation, Stanford University.

Al-Khattat, I.M. (1982) Plane Strain Upsetting: A New Approach to the General Contact Problem in Mechanics. Numerical Methods in Industrial Forming Processes, Proc. Int. Conf., Swansea, UK, Eds. Pittman, J.F.T., Wood, R.D., Alexander, J.M., and Zienkiewicz, O.C. pp 109-122.

Bridgman, P.W. (1952) Studies in Large Plastic Flow and Fracture, 1st Edition, McGraw-Hill.

Goodman, L.E. (1980) Contributions of Continuum Mechanics to the Analysis of the Sliding of Unlubricated Solids. Solid Contact and Lubrication, Eds. Cheng, H.S., and Keer, L.M., ASME, AMD-Vol 39, pp 1-12.

Lu, C.Y., Appleby, E.J., Rao, R.S., Devenpeck, M.L., and Richmond, O. (1982) A Numerical Solution of Strip Drawing Employing Measured Boundary Conditions Obtained with Transparent Sapphire. Numerical Methods in Industrial Forming Processes, Proc. Int. Conf., Swansea, UK, Eds. Pittman, J.F.T., Wood, R.D., Alexander, J.M., and Zienkiewicz, O.D., pp 735-746.

Male, A.T., and Cockcroft, M.G. (1964) A Method for the Determination of the Coefficient of Friction of Metals under Conditions of Bulk Plastic Deformation. J. Inst. Metals, 93, pp 38-46.

Male, A.T., and Depierre, V. (1970) The Validity of Mathematical Solutions for Determining Friction from the Ring Compression Test. Trans. ASME, J. Lub. Tech., 92, pp 389-397.

McMeeking, R.M., and Rice, J.R. (1975) Finite-Element Formulations for Problems of Large Elastic-Plastic Deformation. Int. J. Solids Structures, 11, pp 601-616.

Nagamatsu, A., Murota, T., and Jimma, T. (1972) On the Non-Uniform Deformation of Material in Axially Symmetric Compression Caused by Friction. Part 3: Theoretical Analysis of Compression of Hollow Cylinder by Finite Element Method. Bull. JSME, 15, No. 89, pp 1339-1347.

Nagamatsu, A., Murota, T., and Jimma, T. (1970) On the Non-Uniform Deformation of Block in Plane-Strain Compression Caused by Friction. Part 2: Experiments and Proposition of Relative Slip Ratio. Bull. JSME, 13, No. 66, pp 1396-1402.

Sampson, R.J. (1978) Surface II Graphics System (Revision One), No. One, Series on Spacial Analysis, Kansas Geological Survey.

AUTOMATIC GENERATION OF HYBRID AND DISPLACEMENT-TYPE FINITE ELEMENTS

H Alaylıoğlu

NMERI, CSIR, South Africa

INTRODUCTION

An algorithmic scheme is presented for the automatic generation of the stiffness and mass matrices for the hybrid and displacement-type finite elements without the need for numerical integration.

The interior strain and displacement fields for the displacement model and the interior stress and displacement fields, as well as boundary displacements and tractions, for the hybrid model are represented by multi-dimensional real and integer arrays. The multi-dimensional array representation of the finite element properties, in terms of the normalized finite element co-ordinates, are exploited for the algorithmic representation of the finite element matrices. The necessary volume, area and line integrals for the finite elements are computed analytically by proper evaluation of the generated algorithms within simple DO loops of the element generator computer programs.

The presented scheme, with the automated analytical integration facility, is particularly useful for the generation of the hybrid-type finite elements which were reported to be more accurate but computationally more expensive than the displacement-type finite elements. Computational experiences have shown that, for both types of finite element models, the bulk of the computing time is taken up by the numerical integration of the stiffness and mass matrices. The algorithmic representation of the finite element properties and the automated analytical integration facility presented in this paper drastically reduce the computational efforts for the generation of the hybrid and displacement-type finite elements in general-purpose finite element programs.

HYBRID-TYPE FINITE ELEMENT FORMULATION

The formulation is based on the modified hybrid-stress finite

element technique, where an equilibrium stress field and a compatible displacement field are assumed independently at the interior as well as along the boundaries of each element. The method makes use of the Hellinger-Reissner energy principle (see Pian (1973)). In the usual hybrid-type finite element formulation, a vast amount of algebra is required for the exact evaluation of the integrals and, because of this drawback, numerical integration is preferred. In general, numerical integration procedures are inherently costly and may cause some difficulties in the matrix inversions due to the loss of accuracy in the integration. These deficiencies are usually more pronounced in the analysis of large order structures. In the proposed hybrid-type finite element formulation, all the necessary integrations are performed analytically with the use of computerized symbolic manipulations. For this purpose, the stress and displacement variables at the interior and along the edges of the hybrid elements are represented by multi-dimensional arrays.

Symbolic representation of the element variables

For the representation of the interior stress field, each component of the interior stress resultants (forces and moments) F_i can be expressed by a two-dimensional array with m unknown stress coefficients β.

$$F_i = \Omega_{im} \beta_m \tag{1}$$

where Ω_{im} is the shape function associated with each β, the index i indicates the particular component of the stress field and as is a general rule in index notation, repeated indices denote summation over the range. Each of the shape function components Ω_{im} can be expressed in terms of the normalized coordinate variables η and ξ with their exponents A_{im} and B_{im}, and the associated coefficients P_{im}.

$$\Omega_{im} = P_{im} \eta^{A_{im}} \xi^{B_{im}} \tag{2}$$

The interior stress field components given by Equation (1) should satisfy the equilibrium conditions applicable to the problem. Consequently, the equilibrium dependent stress coefficients need to be expressed in terms of the independent coefficients, and these independent coefficients can appear more than once in the stress field. Because of this cross-coupling between the components of the stress field and/or inter-coupling within the particular stress field component, the shape function Ω_{im} given by Equation (2) generally assumes the form

$$\Omega_{im} = P_{im}\, \eta^{A_{im}}\, \xi^{B_{im}} + P^{*}_{im}\, \eta^{A^{*}_{im}}\, \xi^{B^{*}_{im}} \tag{3}$$

where the additional terms with the asterisk represent the cross and/or inter-coupling effects. An independently assumed displacement field within each element can be expressed in a similar form.

$$U_i = \Lambda_{if}\, q_f \tag{4}$$

where U_i is the i th component of the interior displacement field, q_f is the generalized nodal displacement and f is the number of degrees of freedom of an element. Λ_{im} is the shape function component which is expressed as

$$\Lambda_{if} = R_{ifc}\, \eta^{C_{ifc}}\, S_{ifd}\, \xi^{D_{ifd}} \tag{5}$$

in terms of the shape function coefficients R_{ifc} and S_{ifd} associated with the normalized co-ordinate variables η and ξ, and their exponents C_{ifc} and D_{ifd}. The indices c and d designate the number of entities within each of the shape function terms.

In the hybrid-stress technique, the boundary displacements U_{rs} along the edges of the element need to be formed separately. These displacements should be compatible with the displacements within the element as well as with the displacements of the neighbouring elements. Thus, being compatible with the respective shape functions Λ_{if} of the interior displacements U_i, the shape functions Γ_{rsf} for each q_f of the boundary displacements U_{rs} will assume the following forms:

$$\Gamma_{rsf} = G_{rsfc}\, \eta^{I_{rsfc}} \quad \text{or} \quad \Gamma_{rsf} = G_{rsfc}\, \xi^{I_{rsfc}} \tag{6}$$

depending upon the co-ordinates of the boundary. G_{rsfc} is the shape function coefficient associated with either R_{ifc} or S_{ifd} of Equation (5), I_{rsfc} is the exponent of η or ξ and the indices s and r indicate the number of element boundaries and the number of displacement components at each boundary respectively.

Parallel to the formation of the boundary displacements U_{rs} from the interior displacements U_i, the boundary tractions T_{rs}

can be formed from the interior stress resultants F_i of the element. Thus, compatible with the respective shape functions Ω_{im} of F_i, the shape functions T_{rsm} for each β_m of the boundary tractions T_{rs} will assume the following forms:

$$T_{rsm} = H_{rsm}\, \eta^{J_{rsm}} \quad \text{or} \quad T_{rsm} = H_{rsm}\, \xi^{J_{rsm}} \tag{7}$$

similar to Equation (6), where H_{rsm} is the shape function coefficient in conjunction with P_{im} and/or P^*_{im} of Equation (3) and J_{rsm} is the exponent of η or ξ.

Finite element equations

For the formation of the finite element equations, the independently assumed finite element displacements and stresses are generally inter-connected by using the variational principles of elasticity. The modified form of the Hellinger-Reissner principle is used in the present hybrid formulation.

It can be shown that (see Pian (1973)), if the displacements U_{rs} along the interelement boundaries of the element are continuous and the boundary tractions T_{rs} are compatible with the interior stress field, the Hellinger-Reissner energy functional Π_{HR} can be written in the following modified form with the addition of a kinetic energy term for an elastodynamic analysis:

$$\begin{aligned}\Pi_{HR} = \sum_e \Big\{ & \int_{A_e} \left(\tfrac{1}{2} F_i E_{ij} F_j + F_{i(j),j} U_i + \bar{F}_i U_i\right) dA_e \\ & + \tfrac{1}{2}\rho \int_{V_e} \dot{U}_i \dot{U}_i \, dV_e - \int_{S_{s_e}} T_{rs} U_{rs} \, dS_{s_e} + \int_{S_{F_e}} \bar{T}_{rs} U_{rs} \, dS_{F_e} \Big\}\end{aligned} \tag{8}$$

where the summation extends over all elements e, E_{ij} is the elastic compliance tensor, $\bar{F}_i$ is the prescribed body force, V_e is the elemental volume, A_e is the elemental area, S_{s_e} is the interelemental boundary, S_{F_e} is the portion of the boundary on which tractions $\bar{T}_{rs}$ are prescribed, ρ is the material density, and $\dot{U}_i$ is the velocity field component which assumes the same shape as the corresponding displacement component, i.e.,

$$\dot{U}_i = \Lambda_{if}\, \dot{q}_f \tag{9}$$

$\dot{q}_f$ is the first derivative of q_f with respect to time.

The hybrid functional Π_{HR} can explicitly be evaluated by substituting the finite element variables symbolically expressed by Equations (1-7) and (9) for an element as follows:

$$\Pi_{HR} = \frac{1}{2}\{\beta_m \int_{A_e} \Omega_{im} E_{ij} \Omega_{jn} dA_e \beta_n + \rho \dot{q}_f \int_{V_e} \Lambda_{ig} \Lambda_{if} dV_e \dot{q}_g\}$$

$$- \beta_m \int_{S_{s_e}} T_{rsm} \Gamma_{rsf} dS_{s_e} q_f \qquad (10)$$

where j=i, n=m and g=f. The functional Π_{HR} should assume a stationary (minimum) value if the finite element is to remain in equilibrium. The independent field variables for the present hybrid element are the stress coefficients β_m (β_n) and the generalized nodal displacements q_f (q_g). Thus the functional Π_{HR} should be varied with respect to these variables. The first variation of Π_{HR} with respect to β_m, i.e. $\delta\Pi_{HR}/\delta\beta_m=0$, leads to the equation

$$\int_{A_e} \Omega_{im} E_{ij} \Omega_{jn} dA_e \beta_n - \int_{S_{s_e}} T_{rsm} \Gamma_{rsf} dS_{s_e} q_f = 0 \qquad (11)$$

which can be represented in the usual matrix notation as

$$[N]\{\beta\} - [L]\{q\} = 0 \qquad (12)$$

With the use of the notation of Equation (12), Equation (10) can be rewritten as

$$\Pi_{HR} = \frac{1}{2}(\{\beta\}^T[N]\{\beta\} + \rho\{\dot{q}\}^T \int_{V_e} [\Lambda]^T[\Lambda] dV_e \{\dot{q}\}) - \{\beta\}^T[L]\{q\} \qquad (13)$$

Furthermore, by solving for $[N]\{\beta\}$ and $\{\beta\}^T$ from Equation (12) and substituting into above, the functional

$$\Pi_{HR} = -\frac{1}{2}(\{q\}^T[L]^T[N]^{-1}[L]\{q\} - \rho\{\dot{q}\}^T \int_{V_e}[\Lambda]^T[\Lambda] dV_e \{\dot{q}\}) \qquad (14)$$

becomes expressible in terms of one of the independent field

variables, i.e., the generalized nodal displacements $\{q\}$ and their dependent derivatives $\{\dot{q}\}$. The variation of Π_{HR} with respect to these variables can be obtained by applying Hamilton's variational principle (e.g. see Hildebrand (1965)). Thus, in order to attain the stationary value of Π_{HR} at dynamic equilibrium between the time interval of $t_1 - t_2$, the condition of

$$\int_{t_1}^{t_2}(\{\delta\dot{q}\}^T \rho \int_{V_e} [\Lambda]^T[\Lambda]dV_e \{\dot{q}\} - \{\delta q\}^T [L]^T [N]^{-1}[L]\{q\})dt = 0 \quad (15)$$

must be satisfied. If the first term of the above equation is integrated over the interval $t_1 - t_2$, and the result is substituted back into the same equation after satisfying the Hamilton's condition

$$-\int_{t_1}^{t_2}\{\delta q\}^T(\rho \int_{V_e} [\Lambda]^T[\Lambda]dV_e \{\ddot{q}\} + [L]^T[N]^{-1}[L]\{q\})dt = 0 \quad (16)$$

Since $\{\delta q\}$ can assume any value at the nodal points of the finite element, the remaining terms inside the brackets should be equal to zero, i.e.

$$[M]\{\ddot{q}\} + [K]\{q\} = 0 \quad (17)$$

in which

$$[K] = [L]^T[N]^{-1}[L] \quad (18)$$

is the stiffness matrix and

$$[M] = \rho \int_{V_e} [\Lambda]^T[\Lambda]dV_e \quad (19)$$

is the mass matrix of a finite element.

Automatic generation of the stiffness matrix

For the generation of the stiffness matrix of an hybrid element, because of the complexity of the expressions in the triple matrix product of the type given by Equation (18), numerical integration procedures are usually adopted in most of the hybrid formulations published in the literature. Numerical integra-

tion techniques require the evaluation of the integrand given by Equation (18) at specified quadrature points. This increases the computational expenses significantly. As stated recently by Spilker and Singh (1982), much of the computation time in computing the stiffness matrix $[K]$ corresponds to the numerical integration of $[L]$ and $[N]$ matrices.

In the present work, with the use of the index notation for the generation of the characteristic arrays of a hybrid element, a simple algorithm is formed for the automatic generation of a stiffness matrix without numerical integration. In order to obtain an explicit algorithm for the stiffness matrix $[K]$, Equation (11) can be evaluated for a general stress field with six components. Accordingly, if the indices i and j assume values from one to six in conjunction with the non-zero values of E_{ij}, Equation (11) can be written in the following explicit form:

$$\int_{A_e} \{E_{11}\,\Omega_{1m}\,\Omega_{1n} + E_{12}(\Omega_{1m}\,\Omega_{2n} + \Omega_{1n}\,\Omega_{2m}) + E_{22}\,\Omega_{2m}\,\Omega_{2n}$$

$$+ E_{33}\,\Omega_{3m}\,\Omega_{3n} + E_{44}\,\Omega_{4m}\,\Omega_{4n} + E_{45}\,(\Omega_{4m}\,\Omega_{5n} + \Omega_{4n}\,\Omega_{5m}) \qquad (20)$$

$$+ E_{55}\,\Omega_{5m}\,\Omega_{5n} + E_{66}\,\Omega_{6m}\,\Omega_{6n}\}\,dA_e\,\beta_n - \int_{S_{s_e}} T_{rsm}\,\Gamma_{rsf}\,dS_{s_e}\,q_f = 0$$

where the equalities of $E_{12}=E_{21}$ and $E_{45}=E_{54}$ are imposed.

Analytical integration In Equation (20) the integrals over the domain of the finite element can be performed analytically after substituting the definitions of the characteristic arrays given by Equations (3), (6) and (7). These arrays are independent of the element geometry since they were written in terms of the normalized variables η and ξ. Thus, for an arbitrarily shaped finite element whose domain is A_e, Equation (20) can be integrated analytically to yield the following:

$$A_e \left\{\frac{E_{11}\,P_{1m}\,P_{1n}}{(A_{1m}+A_{1n}+1)(B_{1m}+B_{1n}+1)} + E_{12}\left\{\frac{P_{1m}\,P_{2n}}{(A_{1m}+A_{2n}+1)(B_{1m}+B_{2n}+1)}\right.\right.$$

$$\left.\left. + \frac{P_{1n}\,P_{2m}}{(A_{1n}+A_{2m}+1)(B_{1n}+B_{2m}+1)}\right\} + \frac{E_{22}\,P_{2m}\,P_{2n}}{(A_{2m}+A_{2n}+1)(B_{2m}+B_{2n}+1)}\right.$$

$$+ \frac{E_{33}\,P_{3m}\,P_{3n}}{(A_{3m}+A_{3n}+1)(B_{3m}+B_{3n}+1)} + \frac{E_{44}\,P_{4m}\,P_{4n}}{(A_{4m}+A_{4n}+1)(B_{4m}+B_{4n}+1)}$$

$$+E_{45}\left\{\frac{P_{4m}\,P_{5n}}{(A_{4m}+A_{5n}+1)(B_{4m}+B_{5n}+1)}+\frac{P_{4n}\,P_{5m}}{(A_{4n}+A_{5m}+1)(B_{4n}+B_{5m}+1)}\right\}$$

$$+\frac{E_{55}\,P_{5m}\,P_{5n}}{(A_{5m}+A_{5n}+1)(B_{5m}+B_{5n}+1)}+\frac{E_{66}\,P_{6m}\,P_{6n}}{(A_{6m}+A_{6n}+1)(B_{6m}+B_{6n}+1)}\Big)\beta_n$$

$$-\frac{H_{rsm}\;G_{rsfc}\;q_f}{J_{rsm}+I_{rsfc}+1}\,S_{s_e}=0 \qquad (21)$$

The above algorithm is independent of the element geometry and is written briefly in conjunction with the first part of the stress field shape function Ω_{im} given by Equation (3). The second part of this shape function, where the terms with the asterisk were written to represent the cross and inter-coupling effects within the stress field, should also be considered in the analytical evaluation of the integral. Thus, for a general form of a stress field, the first term of Equation (21) should be replaced by the following terms:

$$E_{11}\left\{\frac{P_{1m}\,P_{1n}}{(A_{1m}+A_{1n}+1)(B_{1m}+B_{1n}+1)}+\frac{P_{1m}\,P^{*}_{1n}}{(A_{1m}+A^{*}_{1n}+1)(B_{1m}+B^{*}_{1n}+1)}\right.$$

$$\left.+\frac{P^{*}_{1m}\,P_{1n}}{(A^{*}_{1m}+A_{1n}+1)(B^{*}_{1m}+B_{1n}+1)}+\frac{P^{*}_{1m}\,P^{*}_{1n}}{(A^{*}_{1m}+A^{*}_{1n}+1)(B^{*}_{1m}+B^{*}_{1n}+1)}\right\} \qquad (22)$$

Similar substitutions need to be carried out for the remaining terms of the first part of Equation (21) in order to account for all the coupling effects within the stress field. The algorithmic expression given by Equation (21) is identical with the matrix expression given by Equation (12). Consequently, according to the associated algorithms for the matrices $[N]$ and $[L]$ in Equation (21), the stiffness matrix $[K]$ of an hybrid element can be formed with the use of Equation (18).

Automatic generation of the mass matrix

The mass matrix of a hybrid element can also be generated in the form of an algorithm which can be processed by a computer automatically, without the use of numerical integration procedures. The derivation of the proposed algorithm for the mass matrix follows a procedure similar to that of the stiffness matrix.

We now consider Equation (19) which defines the mass matrix in a classical form, where the integrand is written in terms of the matrix $[\Lambda]$. Every i th row of the matrix $[\Lambda]$ accommodates

the shape functions assumed for the i th component of the displacement or velocity field. If Λ_{if} and Λ_{ig} represent the elements of $[\Lambda]$ and $[\Lambda]^T$ respectively, and if the symbolic representation of these, as written in Equation (5), are substituted into Equation (19), the element M_{gf} of the mass matrix can be expressed as follows:

$$M_{gf} = \rho \int_{V_e} R_{igh}\, \eta^{C_{igh}}\, S_{igp}\, \xi^{D_{igp}}\, R_{ifc}\, \eta^{C_{ifc}}\, S_{ifd}\, \xi^{D_{ifd}} dV_e \qquad (23)$$

where the indices h=p=c=d.

Analytical integration Since the normalized variables η and ξ vary between 0 and 1, the above integral is independent of the element geometry. For an arbitrary finite element with volume V_e, the integral for M_{gf} can be performed analytically to yield the following result:

$$M_{gf} = \rho\, V_e \frac{R_{igh}\, S_{igp}\, R_{ifc}\, S_{ifd}}{(C_{igh} + C_{ifc} + 1)(D_{igp} + D_{ifd} + 1)} \qquad (24)$$

With the use of this algorithm, the mass matrix of the element can be generated automatically by a computer in a very short time.

DISPLACEMENT-TYPE FINITE ELEMENT FORMULATION

The details of the formulation procedure for the displacement-type finite element was given by Zienkiewicz (1977). The same basic procedure is followed for the present development. With the use of matrix notation, the interior displacements $\{U\}$ and strains $\{\varepsilon\}$ of an element can be expressed as

$$\{U\} = [\Lambda]\{q\} \quad \text{and} \quad \{\varepsilon\} = [\Phi]\{q\} \qquad (25)$$

where the elements of $[\Lambda]$ assume the form given by Equation (5), and $[\Phi]$ is related to $[\Lambda]$ with the differential operator matrix $[\nabla]$ corresponding to the strain-displacement relations, i.e., $[\Phi]=[\nabla][\Lambda]$. Accordingly, the element stiffness matrix $[K]$ is formed as

$$[K] = \int_{V_e} [\Phi]^T [\Xi][\Phi] dV_e \qquad (26)$$

in which $[\Xi]$ is the elasticity matrix.

Automatic generation of the stiffness matrix

Since $[\Phi]=[\nabla][\Lambda]$, the element Φ_{jf} of $[\Phi]$ can be formed from the element Λ_{if} of $[\Lambda]$. Φ_{jf} can thus be obtained from Equation (5) as follows:

$$\Phi_{jf} = X_{jfc}\, \eta^{Z_{jfc}}\, Y_{jfd}\, \xi^{Q_{jfd}} \tag{27}$$

If Φ_{ig} represents the elements of $[\Phi]^T$, then the element K_{gf} of the stiffness matrix $[K]$, given by Equation (26), can be written as

$$K_{gf} = \int_{V_e} X_{igh}\, \eta^{Z_{igh}}\, Y_{igp}\, \xi^{Q_{igp}}\, \Xi_{ij}\, X_{jfc}\, \eta^{Z_{jfc}}\, Y_{jfd}\, \xi^{Q_{jfd}}\, dV_e \tag{28}$$

Analytical integration The above form of K_{gf} can also be integrated analytically in general terms. Thus for an arbitrary element with volume V_e, the integrated form of K_{gf} becomes

$$K_{gf} = V_e\, \frac{X_{igh}\, Y_{igp}\, \Xi_{ij}\, X_{jfc}\, Y_{jfd}}{(Z_{igh} + Z_{jfc} + 1)(Q_{igp} + Q_{jfd} + 1)} \tag{29}$$

This algorithm, which is independent of the element geometry, allows the automatic generation of the stiffness matrix in a very short time. For a particular type of finite element, the above algorithm can be written more explicitly according to the non-zero values of Ξ_{ij}.

Automatic generation of the mass matrix

Mass matrices of the displacement and hybrid-type elements are the same, and the algorithm for the element M_{gf} of the mass matrix was given by Equation (24).

CONCLUDING REMARKS

Algorithms are presented for the automatic generation of the hybrid and displacement-type finite elements without the need for numerical integration.

For the hybrid-type elements, the classical hybrid formulation is modified to provide an automation for the analytical integration of the element stiffness and mass matrices. The cost of numerical integration for computing the hybrid-stiffness matrix is usually higher than that of the displacement-stiffness matrix due to the dual field property of the hybrid model. Because of this major set-back, most of the finite element program libraries were built from displacement models, although the hybrid models generally yield better accuracy. The proposed

algorithm promotes the potential use of the hybrid models in general-purpose finite element programs. The algorithm also facilitates a mechanism for the control of the stress distribution. The originally assigned stress distribution of the hybrid element can easily be changed within the DO loop of the program by changing the DATA card for the required number of stress coefficients.

The proposed algorithms provide a simple, accurate and reliable evaluation of the stiffness and mass matrices for both hybrid and displacement-type finite elements and reduce the CPU time significantly as compared to the finite element evaluations with numerical quadratures.

REFERENCES

Hildebrand, F.B. (1965) Methods of Applied Mathematics. Prentice-Hall.

Pian, T.H.H. (1973) Finite Element Methods by Variational Principles with Relaxed Continuity Requirements. Proc. Int. Conf. Variational Meth. Engng, Ed. Brebbia, C.A. and Tottenham, H., Southampton Un. Press, 3:1-24.

Spilker, R.L. and Singh, S.P. (1982) Three Dimensional Hybrid Stress Isoparametric Quadratic Displacement Elements. Int. J. Num. Meth. Engng, 18:445-465.

Zienkiewicz, O.C. (1977) The Finite Element Method. McGraw-Hill.

DESIGN OF STRUCTURAL CONTINUA BY FINITE ELEMENT ANALYSIS OF EQUILIBRIUM MODELS

E.A.W. Maunder

Department of Engineering Science, University of Exeter.

SUMMARY

Most finite element programs applied to structural analysis use displacement models and treat node displacements as the primary unknowns. However, in many cases, the designer would prefer to work directly in terms of stress-resultants and stress fields which satisfy completely the equilibrium conditions.

In this paper a finite element model is proposed using equilibrium elements within which stress fields are defined in equilibrium with edge stress-resultants. The present model is suitable for plane stress or strain problems in linear elasticity. The edge stress-resultants are classified as basic (normal force, shear force, and moment), and higher order. The higher order stress-resultants supplement the basic class and involve edge stresses which are in self equilibrium. The stress-resultants can be treated as element unknowns, and the model can be analysed using the flexibility method.

A computer program has been developed to implement this method of analysis. The main features of the program are described. These include the incorporation of triangular elements having 6 or 9 degrees of freedom of edge stress-resultants, and the selection of suitable force paths from an interior graph embedded in the model. Algorithms for path selections are based on topological properties of the graph.

A numerical example is presented to demonstrate the feasibility of the method, and to demonstrate its potential to structural design, particularly when appeal can be made to the lower bound theorem of plasticity.

1. INTRODUCTION

In many cases, the structural designer requires information on stresses rather than displacements. Stresses should be in

equilibrium so that load paths can be unambiguously defined and understood in the design. With ductile materials design can make use of the lower bound theorem in plasticity. This requires definition of stress fields in complete equilibrium with the loads, i.e. equilibrium in a 'strong' sense. Displacement models normally ensure overall or 'weak' equilibrium, whereas equilibrium models can ensure equilibrium in the 'strong' sense.

Equilibrium models are not new, e.g. de Veubeke (1965), though they do not appear to have gained popularity in use. Compared to conventional displacement models - the formation of overall system equations from element characteristic matrices can be fairly complicated, which may explain its lack of common use.

Watwood & Hartz (1968) describe in general terms how system equations can be formed from general equilibrium elements based on imposing conditions for edge equilibrium. No explicit account appears to be taken of the statical indeterminacy of the model, and their scheme including Lagrangian multipliers appears to lead to an unnecessarily large number of simultaneous equations. Gallagher & Dhalla (1971) describe another general scheme using equilibrium elements - but the system equations are derived based on Airy stress functions defined within the system. Nodal values of these stress functions (and their derivatives) are taken as the basic unknowns - force boundary conditions (i.e. loads) are applied separately using Lagrangian multipliers again. Again the statical indeterminacy is not explicitly accounted for and the number of equations appears unduly large.

Cassell (1976) develops the flexibility method for finite elements - statical indeterminacy and procedures for selection of corresponding force paths are defined, however force transmissions between elements are via corner nodes. This makes the method suitable for use with conventional displacement elements (the flexibility matrices are obtained from stiffness matrices), but it is unsuitable if "strong" equilibrium is required. In this case equilibrium must be satisfied everywhere including element edges. Imposing equilibrium of equivalent nodal forces does not necessarily guarantee equilibrium along element edges.

Allman (1979) describes in detail a triangular equilibrium element with 4 force parameters per edge, however the system equations are formed as a stiffness method with the 4 corresponding edge displacements as unknowns. The assembly procedure is thereby kept 'simple' - but the number of equations may be large.

In this paper, various ideas from the papers already mentioned are combined together and developed into a flexibility method which it is hoped will be computationally efficient. Although attention is restricted here to two dimensional problems, the principles involved should extend to three dimensional problems.

The method will provide directly information of use to the designer, particularly in preliminary stages when an understanding of the 'flow' of internal forces may be sought.

Element force parameters are taken as edge stress-resultants of polygonal elements. The system equations are formed after recognising the statical indeterminacy of the model, and then defining simple force paths between elements satisfying self-equilibrium and equilibrium with the (edge) loads. The boundary conditions are recognised and applied from the outset to the problem. This flexibility method then comes very close to that already established for frameworks e.g. Maunder (1972) - using graph theory for the assembly of the flexibility equations.

2. EQUILIBRIUM ELEMENTS

The elements being considered in this present paper are two dimensional polygonal elements suitable for plane stress or plane strain problems. It is proposed that equilibriating stress fields within an element are based on distributions in equilibrium with unit stress-resultants on each edge. These stress-resultants are classified either as basic or higher order. The basic class comprise normal and shear forces, and an in-plane bending moment i.e. 3 per edge on all edges. One edge should be considered as a support or 'reaction' edge. Thus a polygonal element with p edges has 3(p-1) independent basic stress-resultants - can can thereby accept all edge loading conditions presented as resultant forces and moments.

The higher order class involves edge stresses which are in self-equilibrium along any single edge. They allow for more complicated loading patterns and element interactions, and they require representative parameters to quantify them. For example, a linearly varying shear stress distribution with no resulting force on the whole edge may be represented by a pair of shear forces H_1 directed towards the centre of the edge. H_1 may be taken as the resulting force on half the edge. This stress-resultant and other different types are illustrated in Figure 1.

If each edge has defined 'h' independent higher order stress-resultants, then the total number of independent stress fields corresponding to the stress-resultants is 3(p-1) + p.h.

Definition of suitable stress fields will not be considered here - except to say that they can be defined for a triangular element (e.g. see Allman (1979)), and hence they exist for general polygonal elements!

3. STATICAL INDETERMINACY AND THE INTERIOR GRAPH

Consider a finite element model containing E polygonal elements. The number of edges p_i for the i^{th} element may vary from element

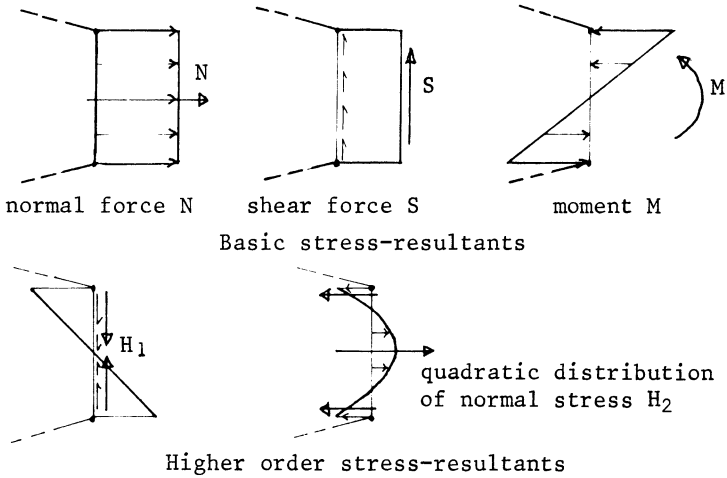


Figure 1 Edge stress-resultants

to element, but let 'h' be constant for every edge.

In the assembled model, element edges may be free (i.e. not connected to anything), or common (i.e. connected to one other element edge or connected to the ground which may be considered as a single rigid element). Let c be the number of common edges.

In order to determine the degree of statical indeterminacy, and to define suitable load and force paths through the structure, it is useful to define within the model a graph. This graph will be termed the interior graph and is defined as follows. Select one interior point in each element, including the ground element if present. These will be the vertices of the interior graph. For each common edge in the finite element model introduce an edge into the graph to connect the vertices corresponding to the two adjacent elements. The interior graph then has E or E+1 vertices, according to whether the ground is included, and c edges.

It can be shown that the degree α of statical indeterminacy is given by

$$\alpha = 3\mu + h\,c \qquad (1)$$

where μ is the cyclomatic number of the interior graph, and complete continuity of the stress vector is assumed across all common edges.

4. SELECTION OF FORCE PATHS FOR THE FLEXIBILITY METHOD

Implementation of the flexibility method requires the formation of α independent self-equilibriating stress fields together with any particular stress field in equilibrium with the applied load. A suitable set of stress fields can now be simply defined based on paths in the interior graph which depend only on the topological properties of this graph.

Each simple cycle (or closed path which does not cross itself) in the graph generates three independent self-equilibriating stress fields involving only the basic class of element edge stress-resultants (or element forces). These correspond to element forces in equilibrium with three bi-actions applied at an arbitrary point termed the cycle origin. The cycle defines the elements stressed by the bi-actions; the bi-actions may be imagined as being introduced into the cycle via a pair of rigid arms and a cut along a common edge. See Figure 2 for an illustration.

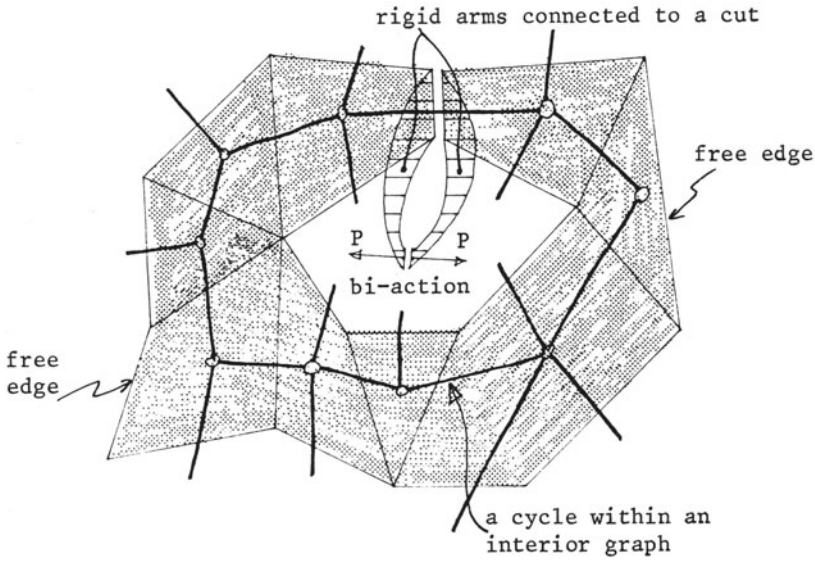


Figure 2 A basic bi-action applied to a cut

Each edge of the graph generates h independent self-equilibriating stress fields involving only the higher order class of element edge stress-resultants. Each such field involves the two elements corresponding to the graph edge and a higher order bi-action applied to the common edges of the elements. See Figure 3 for an illustration.

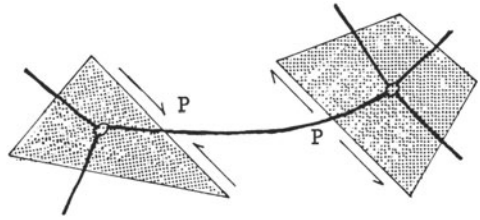


Figure 3 A higher order bi-action

It can be shown that $3\mu + h$ c independent stress fields are thereby generated from selection of μ independent cycles and c edges from the interior graph. This connection between the independence of the two systems is vital to the flexibility method.

For a particular stress field to exist - loads must be applied via element edges as combinations of allowable stress-resultants. A load path through the structure involving only basic stress-resultants can be defined from a spanning tree of the interior graph. The tree would normally be rooted at the vertex corresponding to the ground element if this exists.

Load paths for higher order loads can be limited to those elements to which such load components are directly applied. See Figure 4 for an illustration. The elements stressed by bi-actions or loads in Figures 2, 3, and 4 are shown shaded.

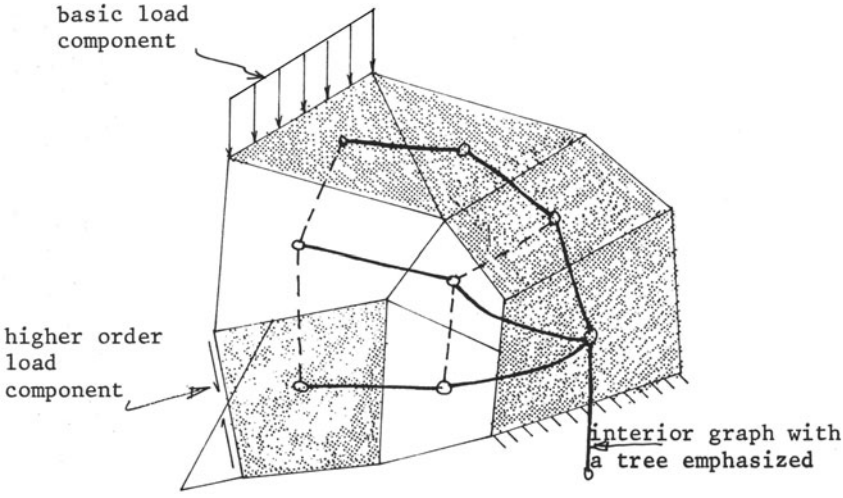


Figure 4 Load paths defined from a tree

5. IMPLEMENTATION OF THE FLEXIBILITY METHOD ON A COMPUTER

5.1 Flexibility equations for the system

Let $\underline{X}$ be the vector of element forces, $\underline{P}$ be the vector of α biactions, $\underline{A}$ be the transformation matrix relating $\underline{X}$ to $\underline{P}$ ($\underline{X} = \underline{A}\,\underline{P}$), $\underline{X}_o$ be a particular vector of element forces in equilibrium with the loads, and $\underline{f}$ be the unassembled flexibility matrix having E element flexibility matrices as submatrices on its diagonal.

The compatibility equations for the system can be written as

$$\underline{F}\,\underline{P} + \underline{V} = 0 \qquad (2)$$

where $\underline{F} = \underline{A}^T\,\underline{f}\,\underline{A}$ and $\underline{V} = \underline{A}^T\,\underline{f}\,\underline{X}_o$.

A computer program in FORTRAN IV has been written to carry out the formation and manipulation of these matrices to achieve a solution for $\underline{P}$ and hence $\underline{X} = \underline{A}\,\underline{P} + \underline{X}_o$.

Problems are of the plane stress or plane strain type and are modelled with triangular elements having h = 1. Thus each element flexibility matrix is 9×9.

5.2 Outline flowchart for the program

An outline of the flowchart appears in Figure 5. The following paragraphs contain brief comments on these main operations.

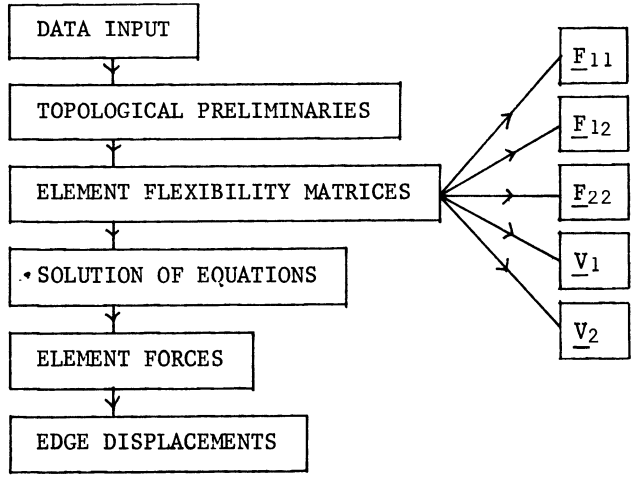


Figure 5 Outline flowchart

5.2.1 Data input The data required is the same as that usually provided to describe a displacement model with three noded triangular elements. However the loads are described as edge loads with up to four components per edge to allow for linear distributions of normal and tangential loading.

5.2.2 Topological preliminaries From the topological data defining element node connections and any supported edges, the computer defines the interior graph as an ordered set of pairs of element numbers. This graph is then used by the computer to select a shortest-route tree, normally rooted in the ground element. This type of tree has the advantage that the least number of elements are involved in any load path used for a particular stress field.

μ independent simple cycles must be selected from the graph. These are chosen by the computer to include all the 'regional' cycles. A regional cycle is considered here as one which passes through a set of elements which all share a common node. For an example, see the shaded elements in Figure 9. These regional cycles are sufficient if the finite element model (including any ground element) contains no holes. If holes are present, additional cycles must be selected to complete the set. Algorithms do exist for this purpose, but present development of the program relies on selection by the engineer and further data input. This is straightforward for two-dimensional models, e.g. insert fictitious elements to fill the holes so that the resulting complex is collapsible (for the topological sense of the term collapsible, see Giblin (1977)), and then select cycles from the interior graph which surround each filling.

5.2.3 Element flexibility matrices The triangular element is split into three triangular subregions so that nine independent cubic stress functions can be defined throughout the element. These functions maintain continuity of the stress vector across subregion boundaries, and hence provide self-equilibriating stress fields which vary linearly in each region. A 9×9 flexibility matrix is formed from these stress functions as described by Allman (1979). This matrix is then transformed by changing the stress field parameters to six basic and three higher order edge stress-resultants as shown in Figure 6.

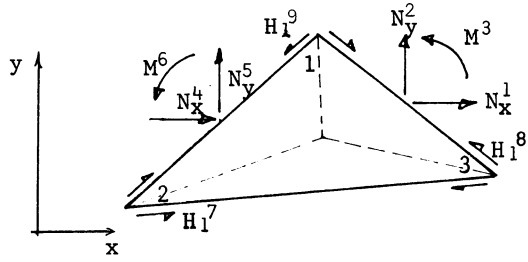


Figure 6 Independent stress-resultants on a triangular element

With reference to Figure 6, stress-resultant type H_1 refers to the shear stress distributions shown in Figure 1. Superscripts refer to the order in which these forces are listed in the 9×1

element force vector. The element flexibility matrix $\underline{f}$ is partitioned into $\begin{bmatrix} \underline{f}_{11} \ (6\times6) & \underline{f}_{12} \ (6\times3) \\ \underline{f}_{21} \ (3\times6) & \underline{f}_{22} \ (3\times3) \end{bmatrix}$ to aid assembly of its contributions to $\underline{F}$. The bi-action matrix $\underline{P}$, the system flexibility matrix $\underline{F}$, and $\underline{V}$ are similarly partitioned to separate the basic from the higher order terms.

Thus $\underline{P} = \begin{bmatrix} \underline{P}_1 \\ \underline{P}_2 \end{bmatrix}$ ← 3μ basic bi-actions
← c higher order bi-actions

and $\underline{F} = \begin{bmatrix} \underline{F}_{11} \ (3\mu\times3\mu) & \underline{F}_{12} \ (3\mu\times c) \\ \underline{F}_{21} & \underline{F}_{22} \ (c\times c) \end{bmatrix}$, $\underline{V} = \begin{bmatrix} \underline{V}_1 \ (3\mu\times1) \\ \underline{V}_2 \ (c\times1) \end{bmatrix}$

The elements are considered one by one, with all the submatrix contributions to $\underline{F}$ and $\underline{V}$ being evaluated and then accumulated in $\underline{F}_{11}$, $\underline{F}_{12}$, $\underline{F}_{22}$, $\underline{V}_1$, and $\underline{V}_2$. $\underline{F}_{21}$ is not required due to the symmetry of $\underline{F}$.

This partitioning of $\underline{f}$, $\underline{F}$, and $\underline{V}$ provides opportunities for options in solving the equations (see para. 5.2.4). It also allows simple means for suppressing the higher order terms if required, and then solving for $\underline{P}_1$ only from

$$\underline{F}_{11}\,\underline{P}_1 + \underline{V}_1 = \underline{0} \qquad (3)$$

5.2.4 Solution of equations The present program uses a NAG library subroutine to solve the full set of equations. This subroutine is based on the Cholesky method, and uses double precision (about 12 significant digits on the department's PRIME 550 computer). Due to the method of partitioning the equations, the option exists to eliminate unknowns $\underline{P}_2$ before solving for $\underline{P}_1$. This may help storage requirements, but requires the inversion of $\underline{F}_{22}$.

Other solution schemes such as the frontal technique can be investigated.

5.2.5 Element forces The element forces $\underline{X}$ (derived from $\underline{P}$ and $\underline{X}_o$) provide immediate information on the 'flow' of forces through the model. Graphical output of the resultants of each set of common edge stress-resultants could provide the design engineer with a useful visual aid. Normal and shear stresses along element edges can be simply found. More details of the linearly varying stress fields within elements may be evaluated if required.

5.2.6 Edge displacements The coefficients in the element flex-

ibility matrices can be considered as element deformations due to unit loads. In particular, the coefficients in $\underline{f_{11}}$ can be interpreted as edge displacements relative to the reaction edge. In the assembled structural model, element forces $\underline{X}$ produce element deformations $\underline{\delta}$. The displacement of any edge relative to a supported edge can be found by summing the influences of the deformations of each element encountered on a path between the two edges. A path which includes the least number of elements is defined by the shortest-route tree of the interior graph.

This part of the analysis has not yet been written into the program, but it is anticipated that edge displacements so found will be directly comparable with those found by the stiffness method of Allman (1979).

7. NUMERICAL EXAMPLE

The program outlined in Section 6 was used to analyse a simple cantilever with an end load. The model and the load distribution are shown in Figure 7. The model has 18 elements and 16 nodes. The interior graph is shown in Figure 8, with the selected tree emphasized by heavy lines. The graph has six regional cycles, two of which are emphasized in Figure 9. The statical indeterminacy = $3\times6 + 1\times24 = 42$.

The analysis produces four stress-resultants for each edge. The results are presented graphically in Figure 10 for the horizontal and vertical edges only. Interaction between elements is shown by a pair of force vectors acting at the quarter points of the edges. Each pair of forces is statically equivalent to the

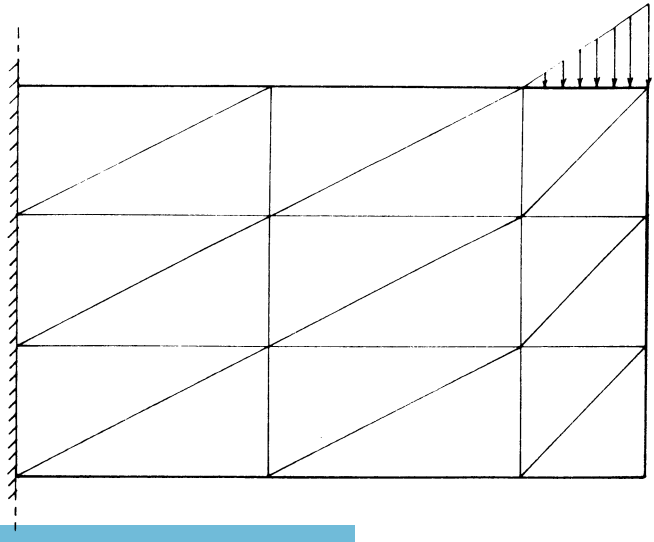


Figure 7 Finite element model and load pattern

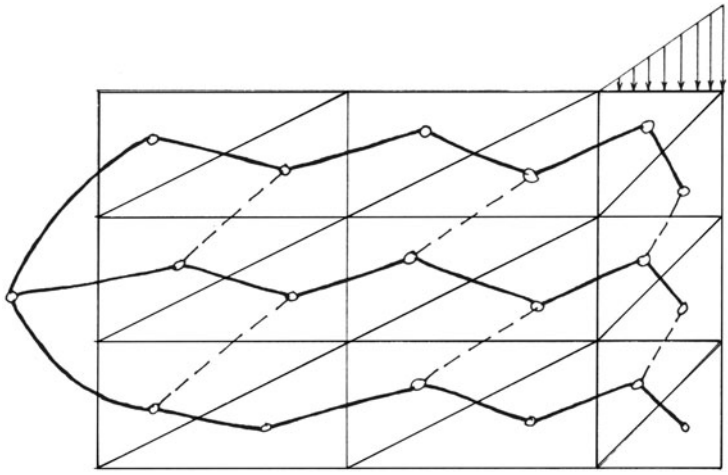


Figure 8 Interior graph with a tree emphasized

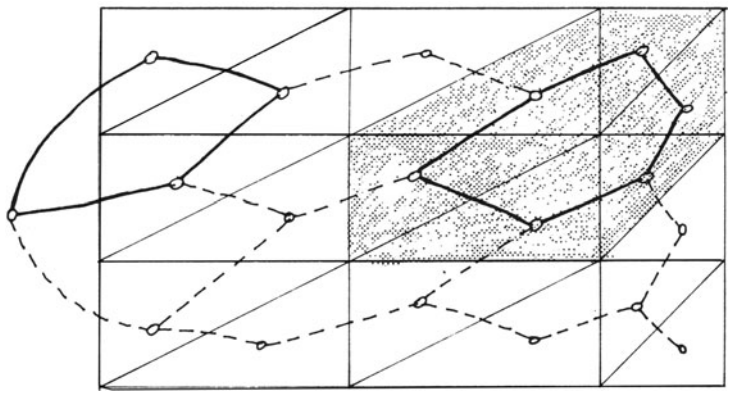


Figure 9 Interior graph with 2 cycles emphasized

four stress-resultants and includes the effects of the higher order stress-resultant H_1. The sense of these forces is given by the action of each element on the 'joints' between it and the neighbouring elements. This form of diagram helps to visualise the flow of forces through the structure in a way similar to that obtained from thrust lines.

Figures 11(a) and 11(b) show distributions of normal and shear stresses respectively along horizontal and vertical edges. When the higher order terms are suppressed (h = 0) these distributions change as shown by the dotted lines. However it can be

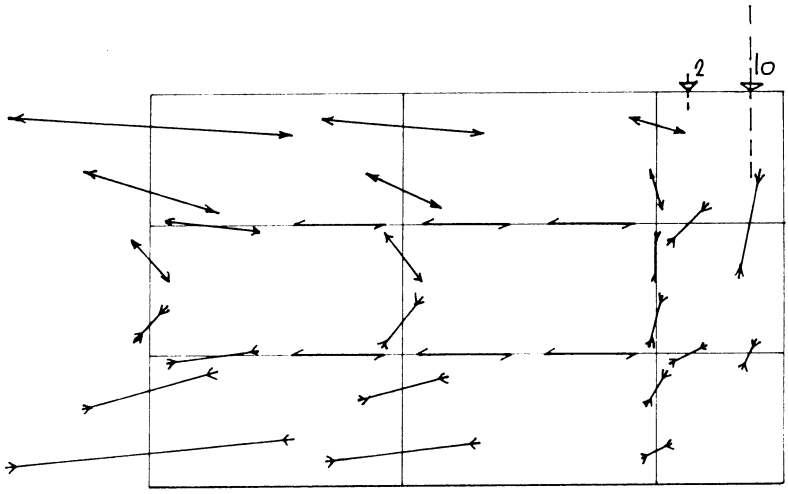


Figure 10 Edge interactions

seen that the average level of edge stress does not change significantly.

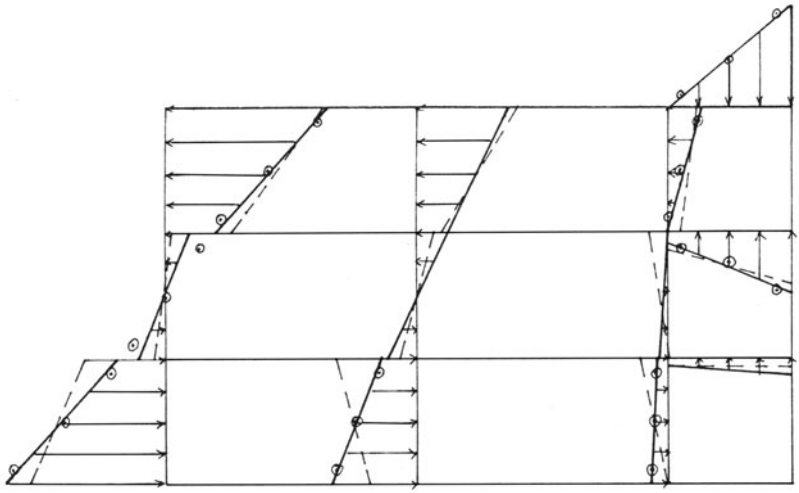
A displacement model was analysed for comparison using 8-noded rectangular elements (having linear strain fields) with edges as shown in Figure 11. Comparing the stresses obtained using the equilibrium model (with h = 1) and the displacement model, there is in general good agreement, particularly with normal stresses.

However differences are apparent with shear stresses; the displacement model implies (by summation of shear stresses) shear forces in excess of the load by about 25%. The equilibrium model makes a good attempt at producing a parabolic distribution

Table 1

Degrees of freedom of some finite element models

Model	Degrees of freedom
Equilibrium, (h = 1)	42
Equilibrium (h = 0)	18
Equilibrium (h = 1), but elements used in a stiffness method with 4 displacements per edge	120
Displacement, 9-8 noded quadrilaterals with 2 displacements per node	66
Displacement, 18-6 noded triangles with 2 displacements per node, and linear strain fields	84

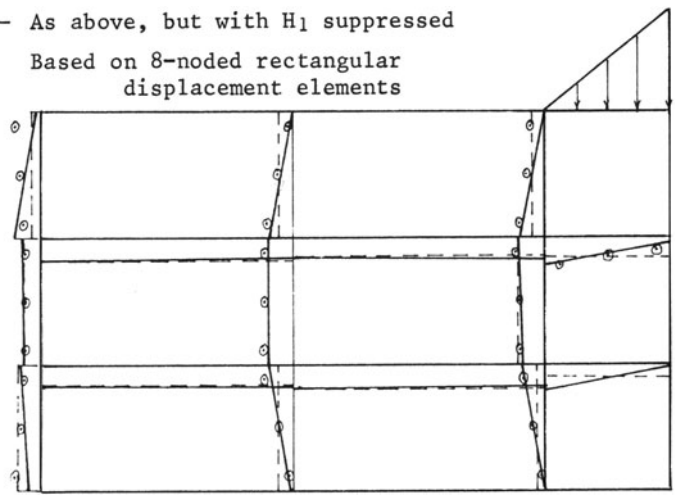


(a) Normal stresses

——— Based on triangular equilibrium elements

– – – As above, but with H_1 suppressed

⊙ Based on 8-noded rectangular displacement elements



(b) Shear stresses

Figure 11 Stress distributions

of shear stress, and of course the total shear force has the correct magnitude.

Comparison of computational efficiencies with different models is not so simple, but it is of interest to compare the degrees of freedom. These are given in Table 1.

8. CONCLUSIONS

(i) For plane stress or strain problems, the flexibility method using equilibrium elements can be as fully automated as the conventional stiffness method. This is essentially due to the topological methods which enable α self-stressing fields to be simply and efficiently defined. These methods only involve integer variables, and the computational effort should not be significant.

It is therefore suggested that 'the final nail in the coffin of the force (or flexibility) method' as described by Davies (1982) has been extracted!

(ii) In terms of degrees of freedom, the flexibility method using equilibrium elements may offer significant advantages over the stiffness method using equilibrium or displacement elements. Such an advantage is even more likely when the higher order stress-resultants are suppressed. A relatively coarse mesh of elements with few degrees of freedom may be quite adequate for design purposes when the lower bound theorem of plasticity is applicable.

(iii) The flexibility method leads to results (in the form of element forces) which can give directly useful information on the 'flow' of internal forces. This sort of information is particularly valuable when thrust lines are required in investigations into the stabilities of masonry structures.

9. ACKNOWLEDGEMENTS

Acknowledgements are due to Mr. J.C. de C. Henderson for his support during the preparation of this paper, and due to Dr. D.J. Allman for computational checks on the results in the numerical example.

REFERENCES

Allman, D.J. (1979) On Compatible and Equilibrium Models with Linear Stresses for Stretching of Elastic Plates. Energy Methods in Finite Element Analysis, Wiley, Chapter 6.

Cassell, A.C. (1976) An alternative method for finite element analysis: a combinatorial approach to the flexibility method for structural continua and to analogous methods in field analysis. Proceedings of the Royal Society, London, A.352, pp. 73-89.

Davies, G.A.O. (1982) Virtual Work in Structural Analysis, Wiley, p.261.

Fraeijs de Veubeke, B. (1965) Displacement and equilibrium models in the finite element method. Stress Analysis, Wiley, Chapter 9.

Gallagher, R.H., Dhalla, A.K. (1971) Direct Flexibility Finite Element Elastoplastic Analysis. Proceedings First International Conference on Structural Mechanics in Nuclear Reactor Technology, Berlin.

Giblin, P.J. (1977) Graphs, Surfaces and Homology, Chapman & Hall, pp.113-118.

Maunder, E.A.W. (1971) Topological and Linear Analysis of Skeletal Structures. Ph.D. Thesis, Civil Engineering Department, Imperial College, London.

Watwood, V.B., Hartz, B.J. (1968) An Equilibrium Stress Field Model for Finite Element Solutions of Two-Dimensional Elastostatic Problems. International Journal of Solids and Structures, Vol.4, pp.857-873.

FINITE ELEMENT ANALYSIS WITH MICRO-COMPUTERS

LE MINH TRI - M. PICCO - F. VAN DEN BURG

Engineers at CETIM'S Computation department

CETIM - 52, Avenue Félix Louat - 60304 Senlis Cedex
France

CETIM

CETIM is the technical center of the French Mechanical engineering industry, and so is a non-profit-making-public utility establishment.

CETIM's Computation Department is subdivised into several units whose activities concern the following fields:

- Mathematical analysis of static or dynamic behaviour of structures in the elastic plastic or viscoplastic domain.

- Structure analysis in conditions which may bring about fracture or which come from malfunctionning of machinery.

Among all these activities which aim to establish a permanent connection between university basic research and industrial applications there is an important part devoted to spreading of knowledge and teaching of new structure dimensionning methods.

According to the progress of technology in data processing with Micro-computers, CETIM started the developpment of general computer programms based on the Finite Elements method for micro-computers.

The aim of the "Methodes de calcul pour les bureaux d' études" service is :

- to develop advanced computing method on micro-computers.

- to supply firm with direct assistance in choice or use of micro-computer.

STRUCTURE ANALYSIS ON MICRO-COMPUTER

Computing the dimensions of a structure to make it resist to stresses within acceptable limits is an arduous task for the design office.

During the last years, two main means at very different levels of difficulty have been made available.

- very simple ones such as strength table,
- sophisticated ones such as finite element analysis often difficult to apply and costly due to the necessity of using powerful computers and larges computing codes.

Nowadays, the progress of technology in date processing with microcomputers has considerably narrowed the gap between those extremes, and permitted the development of very valuable computing tools of acceptable cost for small and middle sized companies.

CETIM has started severals actions in this direction and developed computer software easy to handle and of reasonable cost.

The programs we want to present are making easy the sizing of parts in mechanical engineering. They have been written in a question-answering mode and do not requires knowledge in data-processing. These are programs for finite element analysis on wide-spread microcomputers (HP, Micro with CP/M processing system). They may be put at work in quite a number of areas.

First program has been developed to solve 2D or axisymetrical problems for static conditions.

With it :

- meshing is automatic and the structures are displayed,
- analysis is performed by the finite element methods,
- results and their processing are in form of graphs.

Examples of application cited are straps, flanges,hooks, pressure vessels, supports, spanners, gears, as well as stress concentration and crack analysis.

Second program may solve 3D problems for static and dy-

namic conditions concerning beam elements, plates and shells.

Examples of application are frame-work, piping, pressure vessels, tanks, welded structures.

Programs are modular and they offer following possibilities :

- simplified applications,
- possibility of expending,
- easiness in use.

Such software should give to the small and middle sized companies the possibility to get a powerful and up-to-date computing tool at low cost.

CA.ST.OR B.E.

CA.ST.OR B.E. in French means "CAlcul de STructures sur ORdinateurs pour les Bureaux d'Etudes".
First version of CA.ST.OR B.E. runs on HP 9845,9836 and 9816 with a minimum of 187 K-bytes in core and 2 x 1.2 Mega-bytes on flexible disk.
The language used is a Rom based version of basic HP.

A FORTRAN version is also available for some modules on computer with CP/M operating system.

A complete documentation which is regularly updated and available containing :

- User's manual
- Theorical manual
- Test manual
- Presentation manual

CA.ST.OR B.E. is composed of two parts :

CA.ST.OR B.E. 2D performs thermal and static mechanical analysis of plane or axisymmetrical structure

CA.ST.OR B.E.3D performs static mechanical analysis of three dimensional structures which behaviour is linear and elastic.
The main characteristics of CA.ST.OR B.E. are shown in table 1.

Table 1

	CA.ST.OR BE-2D	CA.ST.OR BE-3D (Framework Analysis)	CA.ST.OR BE-3D (Shell Analysis)
Description	Two dimensional Structure Analysis (Plane and Axisymmetrical)	Three dimensional beam Structure analysis	Three dimensional beam and Shell structure analysis.
Pre-processors	- Automatic meshing - Drawing of structure - optimisation of band-with	- Steel sections modulus (to constitue a file of specified sections) - Interactive procedures to generate the input datas.	- drawing of structure - special automatic meshing for Shells.
Analysis modulus	- Static elasticity - Thermal analysis	- Static elasticity - Dynamic analysis	- Static elasticity - Dynamic analysis
Post-processors	- Drawing of results - Interactive procedure for treatment of the results - group of different post processors, each of them specialised for one type of analysis	- Drawing of structure - Stress analysis - Combination of loading case	- Drawing of structure - Stress analysis

STRUCTURE OF CA.ST.OR BE

The general philosophy of CA.ST.OR BE is based on the fact that to day :

MICROCOMPUTER $\Longrightarrow$ SMALL CORE AVAILABLE

So we will see that we must use special technics for equation solving (it is not reasonable to solve stiffness equations in F.E.M. on a micro-computer with the classical Gauss elimination).

But the main method of saving lies in structuring the program itself. Programms must be chained. After a current programm module has performed the required function, another module is loaded into the memory and continues execution. Data are transmitted by being saved on disk.

However a general program like NASTRAN- ASKA - SAP 5 SESAM ... is impossible to run on a micro-computer, so we often have to limit each program for a specific application : (2D Analysis, 3D Analysis, beam analysis, thermal analysis ...)

Modules corresponding to the basic functions like element stiffness calculation, solver, are the sames.

To the user's level we try to introduce an easy and interactive procedure (automatic meshing, menus ...) that makes possible for a non experimented user of a design office to handle the program in less that one day. The user has only to think about "mechanical problem" and is decharged of computer science.

CA.ST.OR BE-2D

CA.ST.OR BE-2D which performs thermal and static mechanical analysis of plane or axisymmetrical structure is composed of 9000 BASIC instructions and two principal modules : MAIL and SOLVER.

MAIL, the mesh generator, is an important part of CA.ST.OR BE-2D . The structures are meshed with a 6 nodes-iso-parametric triangular elements (Figures 1) which allows linear stresses and parabolic displacements.

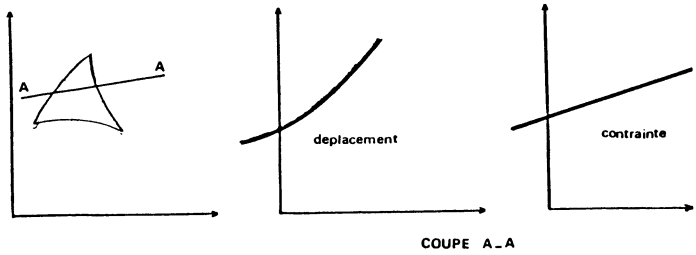


Figure 1

In order to generate the mesh, the description of the boundary of the structure is needed and so the mean sizes of elements.
The numbering of the nodes is optimised in order to minimize computing time in the finite element analysis performed afterwards.
To increase the speed of MAIL two options for the mesh can be used :

- the first option consist in a sample mesh based on the deformation of a rectangle by isoparametric mapping

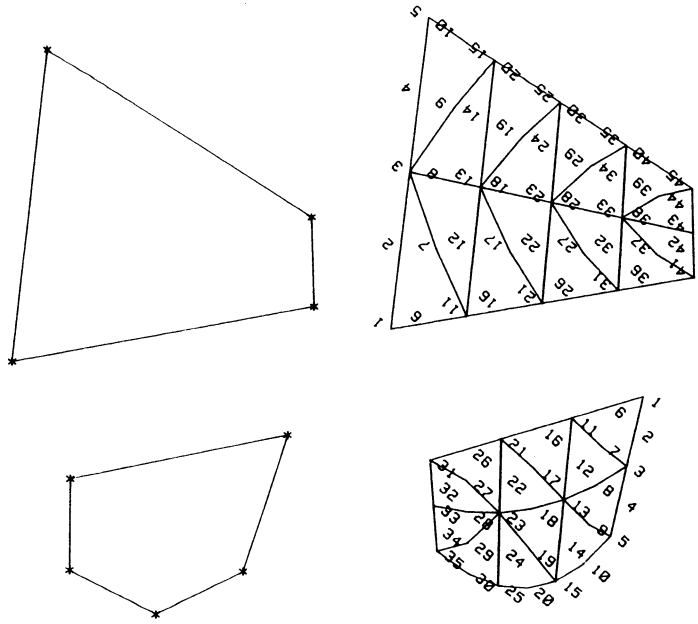


Figure 2 -Deformation of a rectangle

- the second option is more sophisticated and takes more time but it can mesh any kind of piece (figure 3)

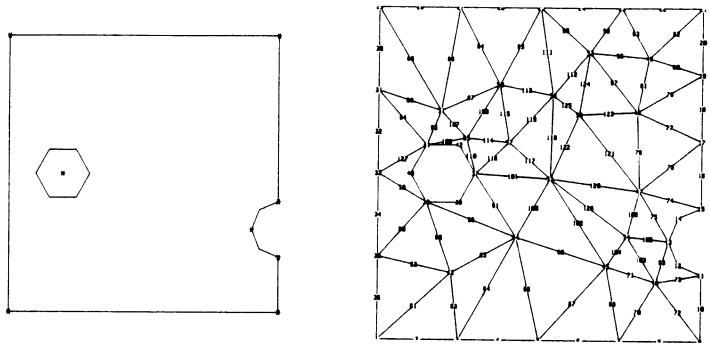


Figure 3

These meshing methods are very simple

- the boundary of the structure is described by an interactive procedure (point, circle, hole...) (fig.4)
- the mean size of the boundary elements is introduced. The algorithm tries to generate elements by layers and so generates a new boundary of the structure (fig.5) The size of the elements is controlled by the user when introducing the boundary. So we must have an algorithm to determine if it is possible to generate an element, if it is not possible we only do a connection (fig. 6)

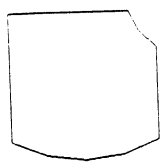


Figure 4
Size of the boundary element is controlled by the user.

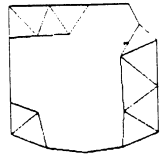
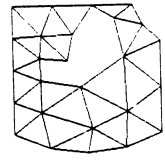


Fig.5
First layer generated.



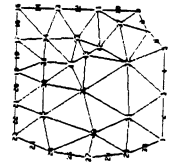


Fig.6
Connections have been done.

This step by step method is very powerful for complex structure but sometimes the mesh is not regular so the operator has the possibility to "move" a node by a sample interactive procedure (Fig.7)

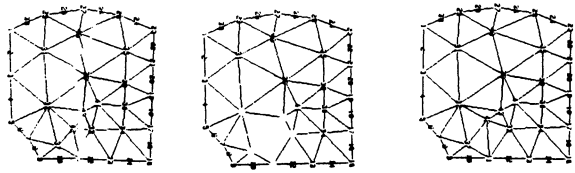


Figure 7

These two options can be used in the same time by dividing into subregions the analysed structure (Figure 8)

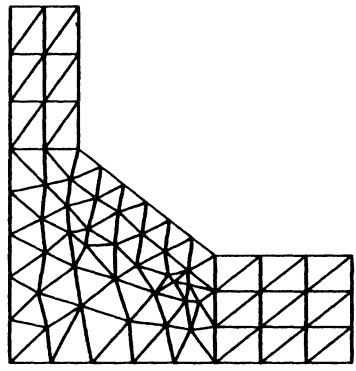


Figure 8

SOLVER

With the increase of high speed computers during the pass decade, the stiffness method has proved to be the most efficient tool in structural analysis. The stiffness method is based on the solution of the equation

$$K \quad U \quad = \quad R$$

with : K : structural stiffness matrix
U : nodal degrees of freedom
R : vector nodal loads

So we have $U = K^{-1} R$

Small computers have a limited core memory capacity (64 - 128 - 187 K-bytes) and the problem to constitute K and find U is very important.
Unless advantage is taken of symmetry and band form of the coefficient matrix the maximum number of equation that can be solved in core is approximately 200 with a 64 K-bytes micro-computer.

However it is possible to increase these possibilities by taking advantage of :

- the elimination of some zero values (skyline method) Figure 9
- the possibility of dividing the matrix into smaller units of convenient sized (guided by the core memory capacity of the computer) Figure 10

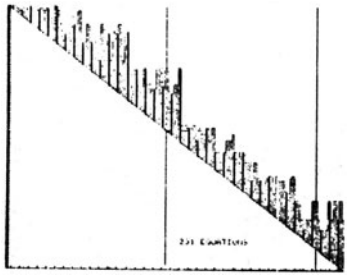


Figure 9

Band Form and Skyline Method

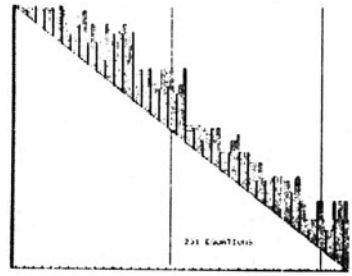


Figure 10

Matrix in blocs
For 187 K-bytes Memory one bloc = 10000 Reals

To constitute and solve the matrix K a skyline method using differents blocs has been implemented.
The size of the blocs is determinated by the capacity of the micro-computer (memory 64 K-bytes $\simeq$ 4000 words)
The size of the problem is determinated by the storage capacity on flexible disk (*)

Regardling the accuracy of calculation we used 12 digits for the equation solver. But we reduced the number of digit using short precision for saving memory for pointer arrays and geometrical data.

USE OF CA.ST.OR BE-2D

CA.ST.OR BE-2D is an interactive program so all data are given by a sample conversational procedure.
The necessary data are :
- description of the geometry (generated by MAIL)
- the conditions on displacements (specified displacements elastics supports ...)
- the loading (concentrated forces, pressure, rotational and gravitational forces ...)

* 187 K-bytes core memory) $\simeq$ 2000 degrees of freedom
2,4 M;bytes flexible disk)

The results are :

- displacements
- stress tensor
- principal stresses and Von Mises and Tresca's criterion at each point of the mesh

Several post processors has been developped for treatment of the results.

For example, the module "COUPE" allows the user to define a cross section in the model and get an evolution of stresses along this cross section. (figures 11 et 12)

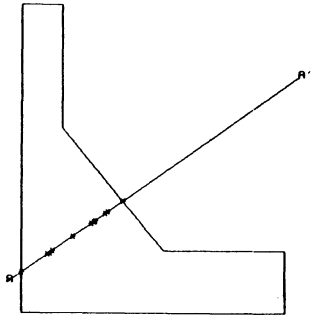


Figure 11
The user defines a cross - section

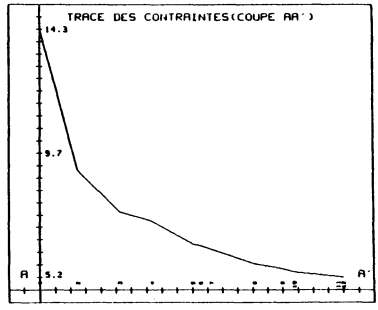


Figure 12
Evolution of the stresses (Tresca) along this section

EXAMPLE : CONNECTING FLANGES

The classical problem of the connecting flanges has been treated using CA.ST.OR BE-2D in about ten minutes to introduce the model and twenty minutes to perform the two dimensional finite element analysis.

We supposed the geometry to be axisymmetric and the flange loaded by an uniform pressure.

A checking has been done by a graphic display of load-case and boundary conditions (figure 13)

Deformation after loading and iso-stresses are the standard results displayed. (figures 14 et 15)

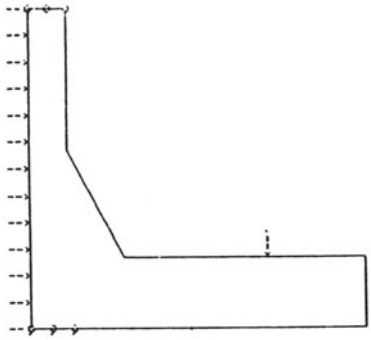


Figure 13

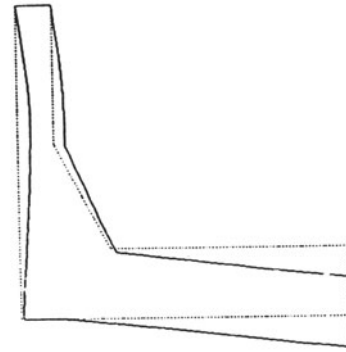


Figure 14

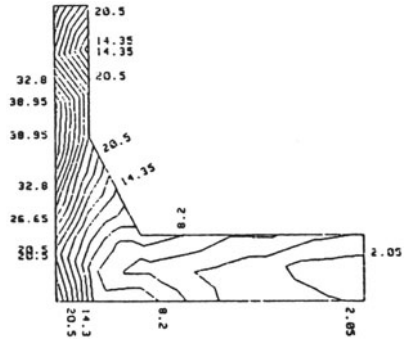


Figure 15

CA.ST.OR BE-3D

CA.ST.OR BE-3D consists of a serie of compatible digital computer programs designed to analyse linear three dimensional elastic structure.

Specific modules dedicated to frame analysis and shell analysis has been implemented in CA.ST.OR BE-3D.

Great emphasis has been placed on making CA.ST.OR BE-3D easy to use.

The engineering user needs very little knowledge of computing or programming. The data formats are clear and concise and arranged as a serie of deck with descriptive titles.

Modules corresponding to the basic fonctions are the same that in CA.ST.OR BE-2D (skiline method using differents blocs for the solver.

FRAME WORK ANALYSIS

A specific module has been developed to analyse frame structures. An interactive procedure helps the user to prepare the input data, geometry, material properties, loads, bondary conditions.

A pre-processor "PROFIL" is used to create a file of special steel sections referenced by a name.

Some classical sections are proposed to the user (Figure 16) and a post-processor calculates stresses in some points of this section (Figure 17)

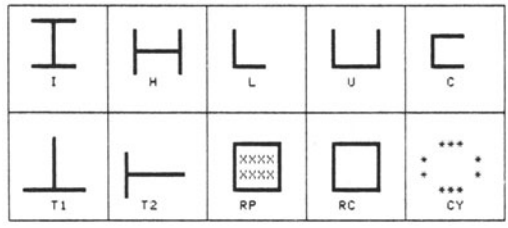


Figure 16

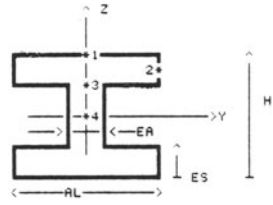


Figure 17

SHELL ANALYSIS

This module is being developped and will be available in June 83 on HP 9845 and HP 9836 - 26 - 16

The module "COQUE" which performs three-dimensional elastic analyses in the assumption of small strains and small displasments, uses a 6 nodes triangular isoparametric element.

The necessary data are :

- the description of the geometry
- the materials properties
- the conditions on displacements (given displacements

elastic support ...)
- the loadings (concentrated forces, tractions on elements ...)

The results are :

- displacements stress tensor, principal stresses and Von Mises and Tresca's Criterion

For the description of the geometry an automatic meshing exists for special structures (Figure 18)

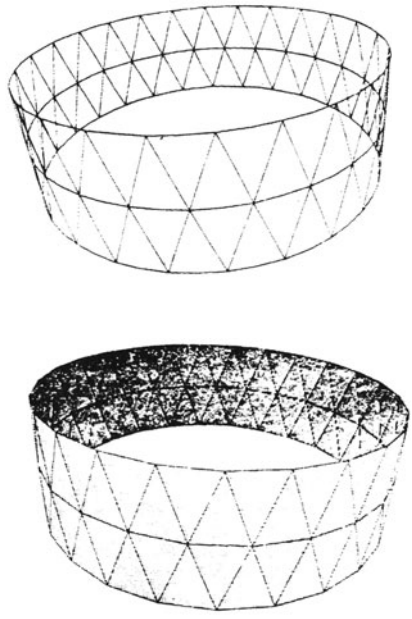


Figure 18

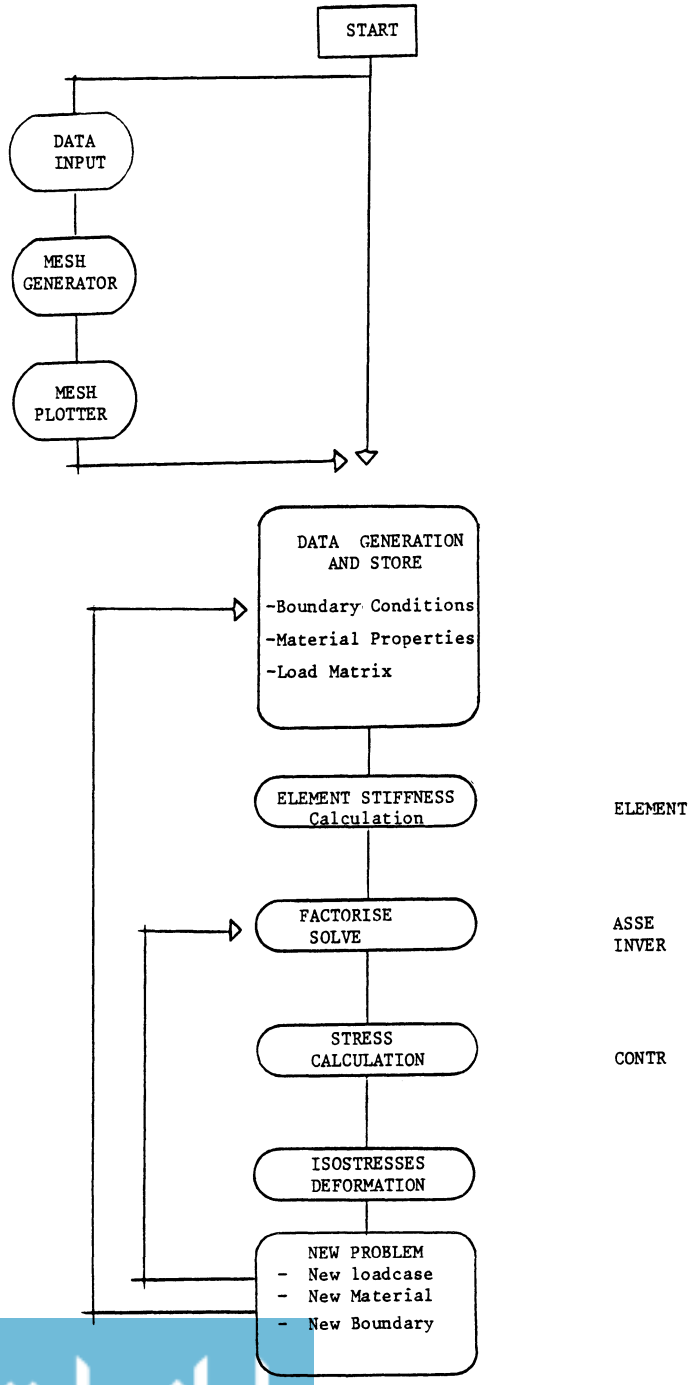
FURTHER DEVELOPMENTS

- CA.ST.OR BE-2D runs on HP 9845, HP 9836

  A Fortran's version will be available in June 1983. This version will run on a 16 bytes Micro-computer with CPM 86 or UNIX operating systems.

- Several automatic meshing for special structures will be implemented (piping analysis) for instance in CA.ST. OR BE-3D and a dynamic computation will be available.

FLOW-CHART OF CA.ST.OR BE-2D



THE FEGS pre-processor FEMGEN

Dr John H Lamont and Dr Geoffrey Butlin

FEGS Ltd 2 All Saints Passage Cambridge England

The FEGS pre-processor FEMGEN

1.1 INTRODUCTION

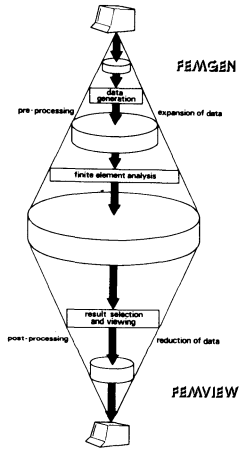
FEMGEN is an interactive graphics pre-processor for the Finite Element Method of analysis. While standard interfaces to the main finite element analysis programs are available from FEGS, FEMGEN has been designed to enable users to write their own interface programs (in FORTRAN) for any finite element analysis program thus providing the user with a common pre-processing facility over the range of his finite element requirements. (FEMGEN is also used as as pre-processor for the Boundary Element Method).

The use of a finite element analysis program comprises essentially three stages:

1 preparation of the model data

2 analysis of the model

3 assessment of the results



Whilst the development of better algorithms and advances in computing technology have enabled finite element techniques to be applied to increasingly more complex problems, the tasks of data preparation and of result assessment can be discouragingly time consuming and prone to error.

FEMGEN provides facilities for the rapid generation of the data representing a finite element model in one of a range of formats to suit the input requirements of most of the major finite element analysis programs.

FEMGEN is based on the top-down method of automatic generation of finite element meshes in which the user defines only the geometric envelope, adjusts mesh parameters interactively from default values, and then inspects the resulting FEMGEN generated mesh on a graphics display.

With its data base specifically designed to give analysis program independence, and its highly organised data structure giving a good interactive response FEMGEN can provide the Finite Element user with an extremely flexible and powerful graphics tool.

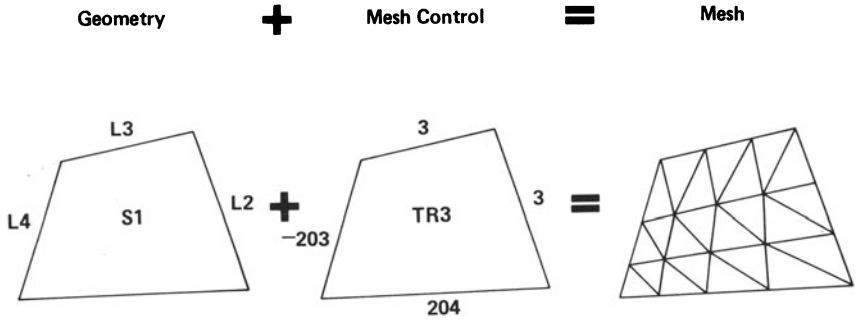
1.2 FEATURE SUMMARY

1.2.1 TOP DOWN

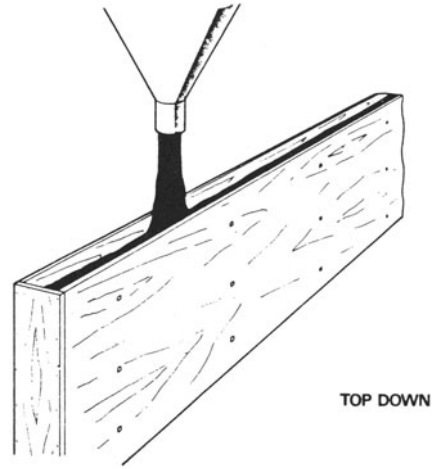
The fundamental data generation facility within FEMGEN is succinctly described in this formula:

Geometry + mesh control = mesh

The user needs only to describe the geometry of the structure, modify default mesh control parameters or add additional ones and then FEMGEN automatically generates the finite element mesh for the structure.



This method of mesh generation is called TOP DOWN and is well illustrated by comparing the generation of a finite element mesh within a model boundary with the pouring of concrete into shuttering - as the shuttering determines the external form of the concrete so the geometrical model in FEMGEN determines the form of the mesh.

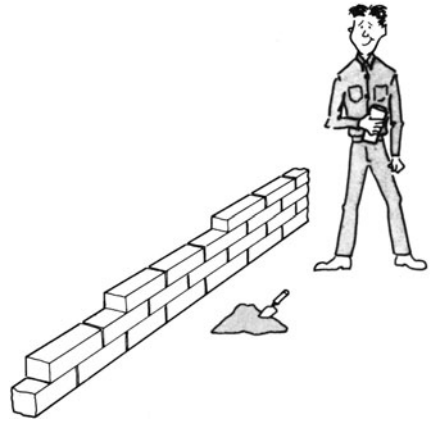


The advantage of the software model over the builders analogy is that the model may be modified and the mesh "poured" again and again until the operator is satisfied with both the external form and the consistency of the mesh.

The advantages of the TOP DOWN method are:

- compact input which implies short lead times and few errors
- powerful editing at the higher geometrical level, eg, if the co-ordinates of a point are changed, the mesh adapts to this change
- easy to control and change mesh density and element types
- constraints, material properties, etc, can be generated with reference to the geometric identifiers
- natural connection to CAD systems

The TOP DOWN method is in strong contrast to the BOTTOM UP method - which is inherent in manual mesh generation procedures and which is the basis for some of the other interactive graphical methods on the market.



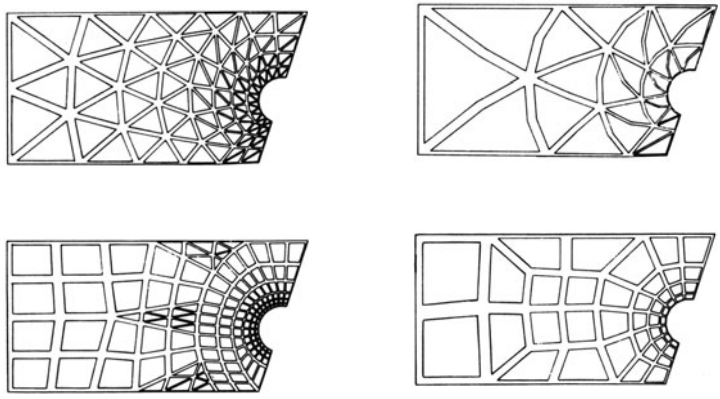
As in the builders' analogy the BOTTOM UP method suffers the disadvantages of:

lack of flexibility
once the mesh has been built;

embedded skills
with little management flexibility;

labour intensive.

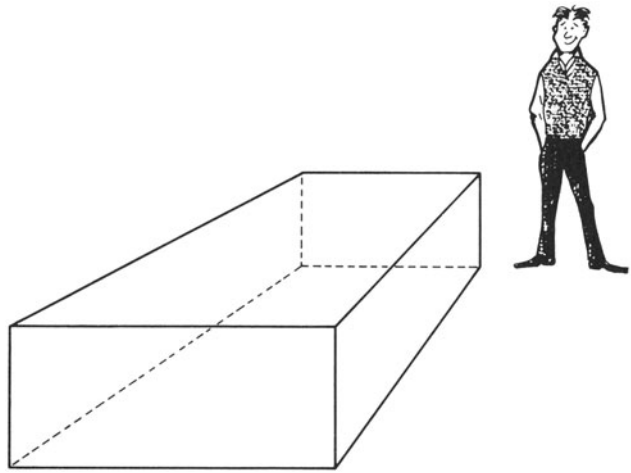
With a range of mesh control parameters specifiable at the interactive graphical screen the engineer can, with a given geometrical profile "steer" FEMGEN to generate a mesh of finite elements with the characteristics he desires.



1.2.2 GEOMETRICAL MODEL

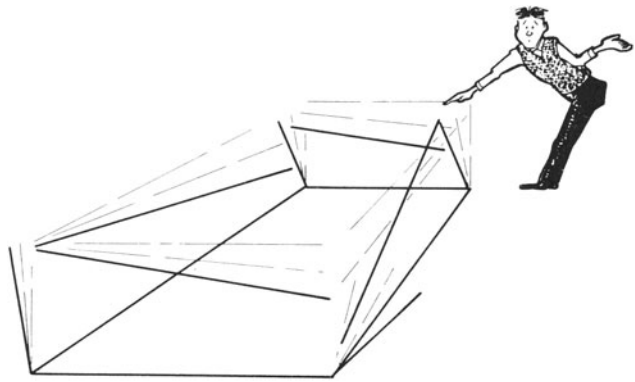
In order to provide such generation facilities under user control FEMGEN has its own solid geometrical model. But before too much is read into that it is necessary to say that this model is specifically designed for finite element mesh generation work and no claim is made that it is of other more general use.

Solid Geometrical Modelling is open to wide interpretation if not misrepresentation. In order to clarify just what solid modelling capability FEMGEN has we consider a computer held model of a simple object shown here

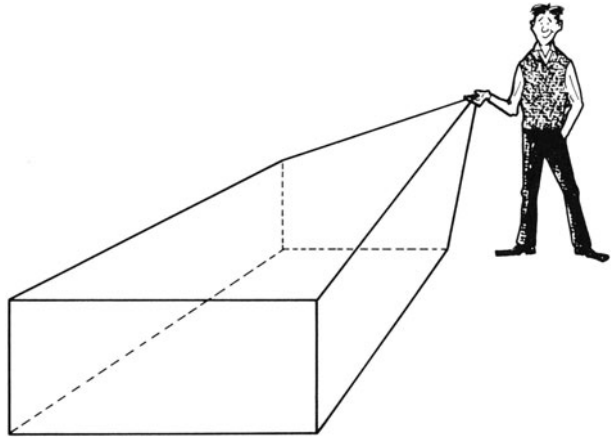


In a computer aided drafting system it can be made to seem very much like the real thing.

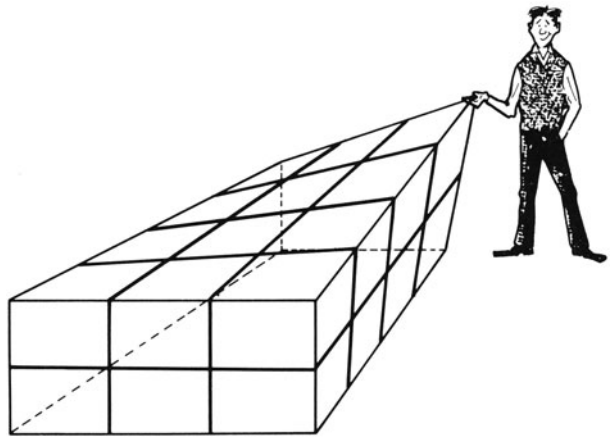
However, when one tries to do something with the model it may, as it were, fall apart. This is not quite so facetious as it may seem since what actually happens is that when one seeks to extract from the model information that appears to be there, one finds that it is in fact not there at all.



Somewhat in contrast the FEMGEN geometrical model is a carefully constructed and maintained hierarchy within which geometric members know their neighbours and the whole model hangs together -

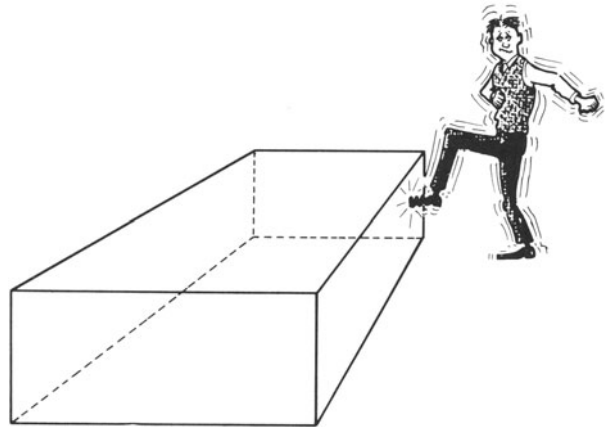


.. in a way which enables meshes to follow the geometric model, that is, lie within and be attached to the geometric model whatever shape it takes.



As emphasised earlier no claim is made that this geometric model provides wide ranging facilities such as some of the solid geometric modellers on the market, it is only designed for finite element mesh generation.

However solid geometric modellers of the top rank (eg, ROMULUS) contain sufficient information to enable the user to know readily whether a point is inside or outside of a defined body.



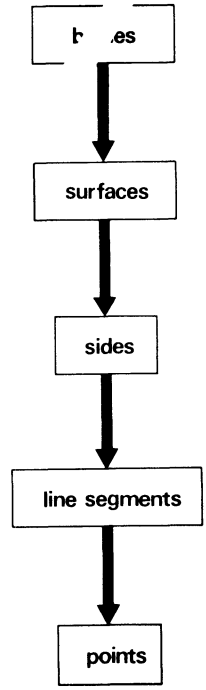
FEMGEN's geometrical model is a hierarchy in which

bodies are defined in terms of surfaces

which are defined in terms of sides

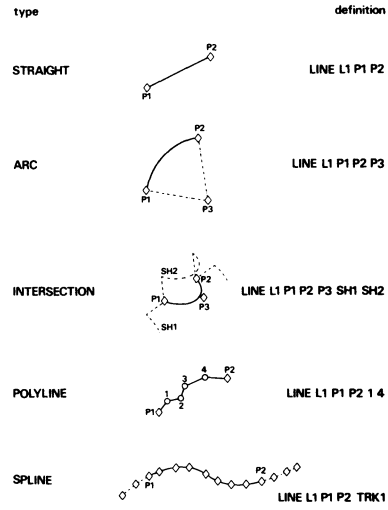
which are defined in terms of lines

which are defined in terms of points and scalars.



All these geometric parts are given names, like point PT1, surface TOP and so on, so that new objects can be defined merely be reference to those defined earlier. The names are chosen by the user, making the interaction with the model natural and convenient. A straight line for example is defined by its start point, eg, P1 and its end point, eg, P2. This means that the only co-ordinates that are needed are those for certain key points. It is, therefore, very easy to modify a geometry by eg, moving a few points and then get a new mesh generated.

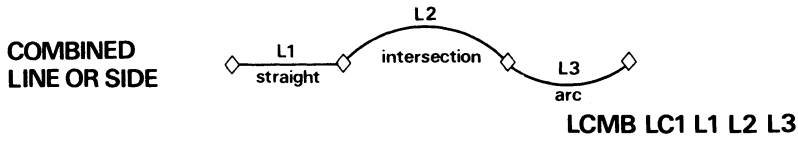
FEMGEN LINES may be of the types shown:



As explained below, the topology and the geometry is treated separately in FEMGEN. This means that the co-ordinates of a point may be changed or the type of a line may be changed by redefining it etc, without affecting the rest of the structure.

It should be emphasized that the calculation of co-ordinates on a line is exact (apart from round off errors), which means that there is no need to artificially split a curved line to avoid approximation errors.

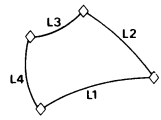
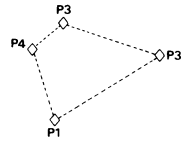
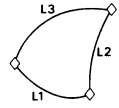
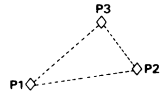
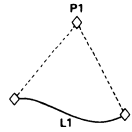
FEMGEN SIDES are either composed of a single line segment or a combination of line segments of any type up to a maximum of four segments a side.



FEMGEN SURFACES Surfaces may be either four sided or three sided and are by default

blended, in which case nodes are mapped onto a shape which is a Coons patch blend between the defining perimeter of line segments making up the sides,

or projected, in which case the nodes are projected onto a shape (sphere, cone, cyliner or plane) "attached" to that surface.

type		definition
BLENDED		
4-SIDED		by lines: SURF S1 L1 L2 L3 L4
		by points: SURF S1 P1 P2 P3 P4
3-SIDED		by lines: SURF S1 L1 L2 L3
		by points: SURF S1 P1 P2 P3
		by point and line : SURF S1 L1 P1

Each of the four sided and three sided surfaces may be defined by the user in terms of either lines or points;

In the case of definition by lines the lines may be either single line segments or combinations of up to four line segments.

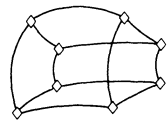
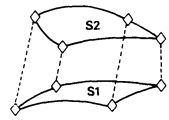
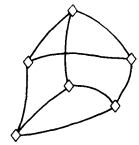
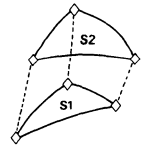
In the case of definition by points FEMGEN will generate the necessary straight lines between the points or if any line segments exist already between the named points then those line segments will be picked up and used in the definition thereby automatically stitching together the geometrical model as the FEMGEN user proceeds to define it - this has special significance for the connectivity of the total mesh: mentioned later.

FEMGEN BODIES may be either six surfaced or five surfaced where generated nodes are placed in space at interpolated positions between the bounding surfaces.

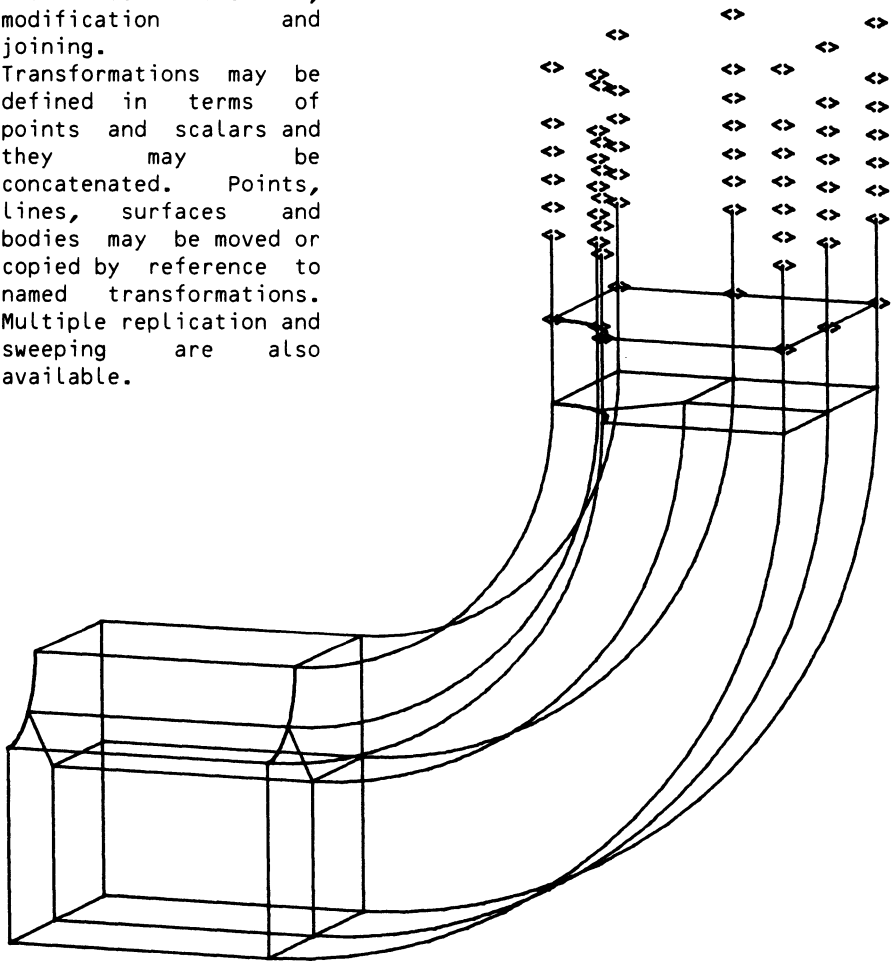
They may be defined

either explicitly by a collection of six or five surfaces,

or they may be defined by only two opposite surfaces in which case FEMGEN will generate the necessary lines and surfaces or will search around to see if line segments and surfaces already exist where they are required and use them if it finds them, again causing a general stitching together of the geometric model being defined by the user.

type		definition
BLENDED **6-SURFACED**		by all six surfaces: **BODY B1 S1 S2 S3 S4 S5 S6**
		by only two opposite surfaces: **BODY B1 S1 S2**
BLENDED **5-SURFACED**		by all five surfaces: **BODY B1 S1 S2 S3 S4 S5**
		by only two opposite surfaces: **BODY B1 S1 S2**

TRANSFORMATION and REPLICATION of geometric parts are available to enhance the fundamental facilities for creation, modification and joining.
Transformations may be defined in terms of points and scalars and they may be concatenated. Points, lines, surfaces and bodies may be moved or copied by reference to named transformations. Multiple replication and sweeping are also available.

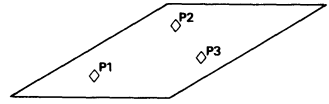


MATHEMATICAL SHAPES are available for:

a) defining the lines intersecting two shapes and points at the intersection of three shapes
and
b) associating with surfaces for the purpose of projecting nodes onto these shapes where this is required.

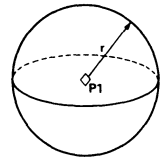
FEMGEN can currently handle the four shapes shown.

PLANE



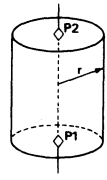
SHPE PL1 P1 P2 P3

SPHERE



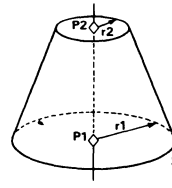
SHPE SPH1 P1 (radius)

CYLINDER



SHPE CYL1 P1 P2 (radius)

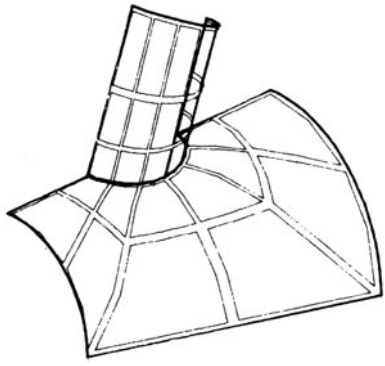
CONE

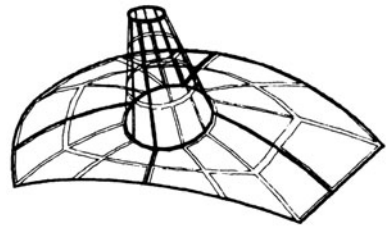


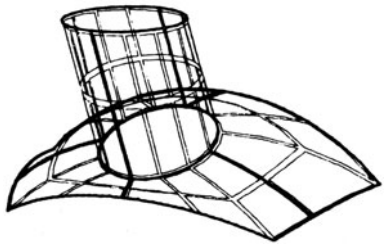
SHPE CON1 P1 P2 (radius)(radius)

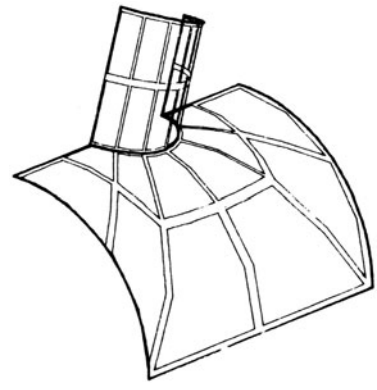
The FEGS pre-processor FEMGEN

Examples of shape intersections calculated by FEMGEN (with meshes):









After mesh generation in FEMGEN the relationships between the geometric parts and the generated ELEMENTS and NODES are retained for the benefit of the user in:

PLOTTING
PRINTING
INTERFACING to finite element analysis programs.

FEMGEN ELEMENTS

FEMGEN elements are geometrically stylised and are mapped, in the finite element analysis interface programs, into the wide range of elements to be found in different analysis programs, thus a four noded element in FEMGEN (QU4) may be mapped into a plane stress element or a plate bending element or a shell element or an exisymmetric element etc, according to the requirements of the analysis.

Element types may be attached in any geometrically compatible configuration (BE's on lines, TR's and QU's on surfaces and PE's and HE's on bodies).

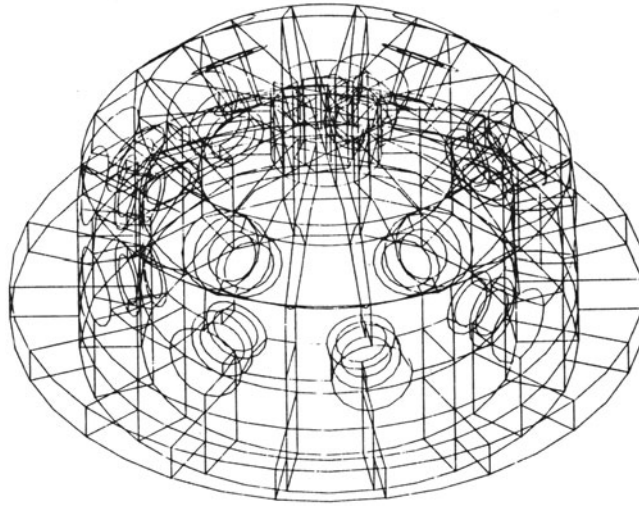
The FEMGEN element library is shown in 1.6 below.

1.2.3 AUTOMATIC MESH GENERATION

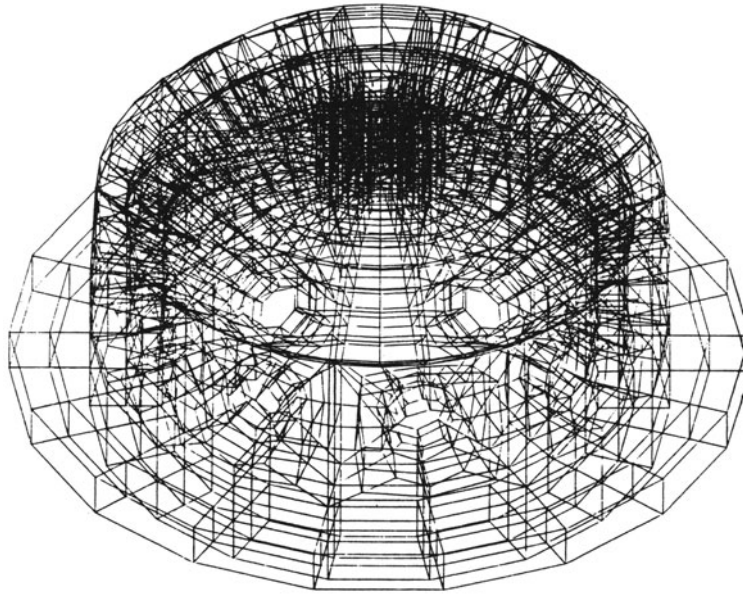
FEMGEN's mesh generation algorithms scan the FEMGEN geometrical model and for each part that has an element type attached to it, FEMGEN generates appropriate numbers of nodes according to division parameters, spacing them according to bias parameters and then generates elements of the type attached to that geometrical part. Furthermore, if a line has beam elements generated within it and that line is a part of a surface which has shell elements within it then these beam elements will be connected via common nodes to the shell elements along that line. Similarly, for beams and solid elements and for shell elements and solid elements.

Thus, as long as the FEMGEN geometrical model is fully connected then the FEMGEN generated mesh will be fully connected. Although it is possible (and occasionally desirable - for example, to impose a crack condition) to define disconnected geometrical models in FEMGEN, FEMGEN 'tries hard' to stitch all geometry together whenever it is requested to generate parts implicitly - for example, when defining a body by giving two opposite surfaces, any exisiting lines or surfaces in between the two indicated surfaces will be found and used in the body to be defined and for a further example, when a request is made to replicate geometry in FEMGEN, FEMGEN will avoid creating any new geometrical part if an existing part of the appropriate type is found to be within a certain tolerance distance of the part to be created (this tolerance distance is under user control).

Example: end cap geometry



and mesh



1.2.4 MANUAL MESH DEFINITION AND EDITING

Because of the dominant top-down approach, FEMGEN's part definition, modification and replication facilities are geared primarily to aiding the user in his definition of a geometrical model rather directly aiding the user in defining a model composed of nodes and elements. The user defines a geometry model and attaches mesh control parameters - FEMGEN makes the node and element model.

However, the user community still demands that FEMGEN provides at least basic facilities for the definition of individual nodes or individual elements or to tweak a mesh here or there. (The increasing supply of geometry to FEMGEN from CAD systems.is increasing the need for a completely top-down approach).

FEMGEN thus currently still provides:

- manual definition of nodes by keyboard command digitizer or graphical cursor;
- definiton of a line as a necklace' of manually defined nodes, which may subsequently be used in a surface definition;
- modification by keyboard or graphical cursor of manually defined nodes or FEMGEN generated nodes;

Graphical cursor controlled node moving is (like graphical cursor controlled point moving) subject to screen depth control according to a given shape ie, nodes (like points) may be 'slid' around FEMGEN shapes.

1.2.5 GRAPHICAL DISPLAY

A vital function of an interactive pre-processor is to make it easy for the user to assess

the appropriateness and accuracy of his geometrical model and

the quality and character of the mesh which has been generated.

The display facilities of FEMGEN's earlier versions were influenced by the dominance of plotter output, hence the PLOT command. However, the graphical display module within FEMGEN has been completely re-written to accommodate the rapidly extending range of display terminals, being bought by users, including raster, colour, beam-refresh and 3-D hardware. (The PLOT command is to be phased into 'DISP' - with an equivalence between PLOT and DISP in this version for old users).

Display commands can add components to a picture from a wide choice of types of attribute such as

geometric perimeter	material property code
part name	physical property code
node and element number	constraint code

for any single part or set of parts.

Beam, shell and solid elements may all be displayed SHRUNKEN.

Once a picture is composed it may be repainted, following a ZOOM or some local change, or sent with a single command to a file in a format appropriate for driving a pen plotter.

On raster displays:

deleted items are erased
modified items are erased and redrawn

On colour displays:

picture components representing objects of different types are resolved with different colours.

On beam refresh displays:

picture components may be erased

and on 3-D hardware:

rotation and depth modulation are supported

1.2.6 COLOUR OPTION

FEMGEN provides the following resolution of picture components by colour:

geometrical perimeters	red
point symbols and names	red
line names, division and bias	yellow
surface names	orange
body names	white
element perimeters and numbers	green
node symbols and numbers	yellow
material and physical property types	blue
constrained nodes	blue

Parts identified with the graphical cursor are confirmed by temporarily changing their colour.

Area fill and shading are avoided for the time being because of the difficulty and expense of getting such images onto paper.

All screen colour is transmitted through into the plotter file for hardcopy production with conventional pen plotters.

1.2.7 GRAPHICAL CURSOR OPTION

A collection of so-called 'Q' or quick commands are provided which use the graphical cursor for their operation.

Command which require mainly
the identification of a geometric part or a mesh part or a non-precise position are candidates for graphical cursor input and include:

defining and moving POINTS

defining, changing, splitting and merging LINES

defining and changing SURFACES

setting DIVISIONS and BIAS

defining and moving NODES

adding and removing parts from SETS

PRINTING of data about a part

Activation of a graphical cursor command is indicated to the user by the display on the screen of the graphical cursor symbol (usually a large plus sign).
Command input with a graphical cursor is by keyboard key hit which causes both the particular key and the screen co-ordinates of the cursor to be sent to FEMGEN.
Most keys have a global meaning such as:

H for HELP producing a display of how the current cursor command may be operated

Identification of parts is made straightforward by FEMGEN 'looking' for the nearest part of the type indicated (by the key hit, or hit code).
When indicating a new position for a point or a node with a graphical cursor two co-ordinates are, of course, readily understood but the depth, into the screen, could easily be ambiguous. FEMGEN solves this problem by enabling the user to invoke cursor depth control by nominating a FEMGEN shape to be used to intersect a line perpendicular to the screen plane though the graphical cursor position; thus points or nodes may be positioned by 'puncturing' a shape or they may be 'slid' around such a shape.

1.2.8 SETS OPTION

FEMGEN SETS provide the user with high level handles with which to manipulate his geometrical model or his mesh.

SETS are simply collections of geometrical parts

SETS are given user names

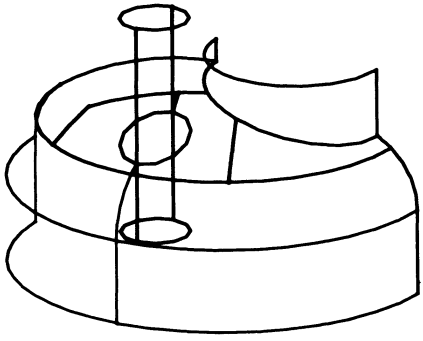
SETS may be given as the argument to many commands including those for:

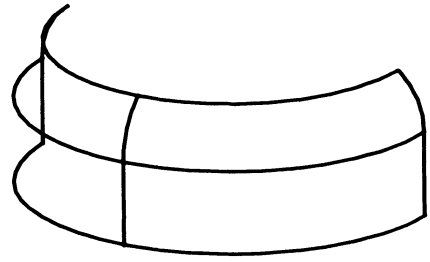
mesh control parameter setting:

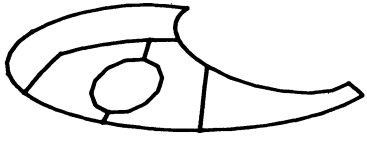
meshing;

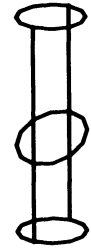
tabulation and display

FEMGEN sets - examples:









1.2.9 NODAL RENUMBERING OPTION

Since FEMGEN follows strict rules for the sequence of the nodes and elements which it generates, it is possible to control these sequences and thus the nature of the matrix of stiffness co-efficients even if this may require in practice dumping the geometrical model to an offline file, editing the order of definitions, reading them back into a new FEMGEN database and then meshing again.

However, this is hardly user friendly and may well prove inadequate in large and/or complex models.

This nodal renumbering option is therefore provided to enable the user to first discover if the nature of the matrix of stiffness coefficients is threatening and if so then to do something about it.

Four renumbering algorithms are provided being those published by Collins Gibbs, Poole and Stockmeyer Grooms and Razzaque

1.2.10 PHYSICAL PROPERTY OPTION

While the main part of FEMGEN provides commands for generating

- node co-ordinates and
- element topologies

this physical property option provides commands for

- associating material and physical properties with elements and
- associating constraints with nodes

Where the user defines material properties, physical properties and constraints on geometrical parts.

As with the node and element generation facilities, these physical property generation facilities are complimented with

- data tabulation and display commands within FEMGEN;
- supported interface programs for a range of finite element analysis programs

1.3 INTERFACING

FEMGEN can be interfaced to other programs via two channels, the COMMAND INPUT LANGUAGE and the LIBRARY OF USER ROUTINES. Both channels are available to the user and are fully documented in the manual.

THE COMMAND INPUT LANGUAGE, the 'front door' of FEMGEN serves two main purposes:

i) driving FEMGEN at the terminal keyboard under INTERACTIVE control;
ii) definition of geometry and mesh control parameters in a pre-prepared OFFLINE FILE;

OFFLINE FILE input, has in turn several applications:

- for preparation of command input prior to reading in batch mode or at the beginning of an interactive session;
- for reading in commands dumped out of FEMGEN for a part of some other model which may be common to the current one;
- for reading in commands generated by user's own Fortran code, eg, when several models are required which can be parameterized;
- for reading in geometry extracted from a computer aided design or drafting (CAD) system.

This last application of offline input is of particular significance and is generally known as CAD/FEM interfacing.

1.3.1 Interfacing to CAD systems

CAD/FEM interfacing has become an issue over the last few years because CAD and FEM are separate established technologies and there is no company providing a single delivery including a comprehensive CAD system as well as a comprehensive FEM system: passing data between these systems is at best a loose linkage and at worst an uncomfortable, tedious and error-prone process.

FEMGEN by virtue of its own, albeit specialised, geometrical model has established iteslf as a bridge between CAD and FEM systems: FEMGEN is interfaced to six commercially available CAD systems on one side and most of the commercially available FEM systems on the other side.

CADAM		ADINA
CAM-X		ANSYS
CDM300	FEMGEN	ASKA
EUCLID		BERSAFE
MEDUSA		MARC
ROMULUS		NASTRAN
+		PAFEC
		SESAM
		STARDYNE
		+

Nevertheless this interface is not yet, and is not likely to be, a 'push-button' affair.

Close co-operation with the producers of CAD systems has highlighted all sorts of problems associated with this transfer of geometric information such as:

- the detail removal problem
- the data pipeline effect
- the users demand for an automatic design/analysis loop
- maintenance responsibility devides between multiple software suppliers
- who drives what

but has also produced some solutions - described in part in Chapter 2, otherwise available as training from FEGS and the suppliers of the above mentioned CAD systems.

1.3.2 Interfacing to Finite Element Analysis programs

The library of user routines is a collection of Fortran routines which provide the means to read data from the FEMGEN database.

The main use of these routines is in the writing of interface programs between FEMGEN and finite element analysis programs. All such interface programs, whether supplied and supported by FEGS, written by a customer or supplied by FEGS and extended by a customer will use these routines and only these routines for reading data out of FEMGEN's data base.

At this version these user routines have been considerably extended, both to encompass the new (physical, material and constraint) data which this version generates and to provide access to many other data items not previously supported by user routines: user routines are documented in Appendix A.

While FEGS suplies and supports a repertoire of interface programs between FEMGEN and the major finite element analysis systems with a varying degree of completeness, all such interface programs

are supplied in source code form at no extra cost;
are written in Fortran
use only the documented user routines for extracting data from FEMGEN's data base;

and are thus readily modified and or extended by a user.

Historically, FEMGEN interface programs to finite element analysis programs have been designed to be operated via a keyboard dialogue, mainly to enable a user to choose which finite-element-analysis-program element-type to map a FEMGEN element type into. However, now that FEMGEN includes the facility for setting element type variants (see 1.6 below) this need has largely disappeared and such interface programs will in future be run mainly in batch mode.

Nevertheless in cases where the whole interfacing is to be packaged in a user friendly way then the interface program will typically contain questions for the user necessary for the automatic preparation of a complete input file for a particular finite element analysis program.

1.4 LIMITATIONS

FEMGEN uses a database on disc to store the entities created during execution. The numbers of entities which can be referenced by FEMGEN is thus dependent on the available disc space.

For internal storage reasons, FEMGEN limits certain other items as follows:

(Number of nodes per line)	—100
(Number of nodes per surface or body) plus (Number of nodes on all intersection lines) where NMAX is set at installation time. Its value may be obtained by using the command PRNT M.	—Nmax
(Biasing value, IQ)	—100
(Number of lines per Side)	— 5
(Number of Picture Segments per Display)	—11
(Set Mechanism Limit)	3,968 each of Points, Lines, Surfaces and Bodies

The command processor places certain restrictions on the format of input data, as follows:

Alphanumeric names must:

a) start with a letter
b) not contain special characters
c) not contain more than 4 characters

Any one command must not include more than 28 parameters

1.5 COMPUTING REQUIREMENTS

FEMGEN is structured in a modular way using a large proportion of machine-independent code. This has enabled FEMGEN to be implemented on a wide variety of hardware having the following facilities:

- 20-60K of 32-bit words of primary memory
- 1 Mbyte of disc memory
- Standard Fortran 66 compiler
- Virtual memory, or load-on-call segmentation facility
  overlaid segments
- Routines for handling characters and direct access disc files

Many general purpose computers meet these requirements, and FEMGEN has been implemented on UNIVAC, CDC, IBM, DEC, DATA GENERAL, PRIME, GEC, PERKIN ELMER, HEWLETT PACKARD and APOLLO machines

The precise computing requirements for the local installation are given in Appendix H.

The modular nature of FEMGEN has enabled it to be interfaced to many graphics devices in a variety of operating environments including:

- batch running with pen plotter
- time-sharing using storage tube terminals
- stand-alone systems using refresh screens

Computer hardware is undergoing rapid technical development, and FEMGEN is carefully designed to be adaptable for new hardware and software environments.

1.6 THE ELEMENT LIBRARY

FEMGEN's elements are stylised geometrically because for the purposes of mesh generation a four-noded plain-strain element is the same as a four-noded axisymmetric element. But in order to provide for the translation of the FEMGEN generated mesh to an input file for a particluar finite element analysis program the FEMGEN user is allowed to assign an element VARIANT to the element TYPE and furthermore, the user is provided with the correlation table between FEMGEN element types/variants and the various types available in each of a range of finite element analusis programs on these lines, eg:

FEMGEN ELEMENT TYPE	variant	ANSYS element
TR3	1	411
	2	423
	3	421
	4	25
	5	63

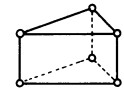
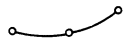
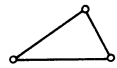
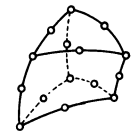
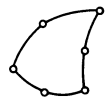
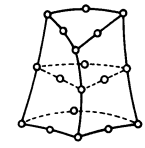
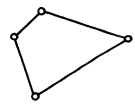
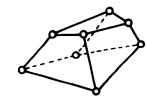
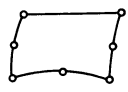
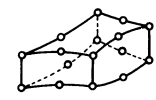
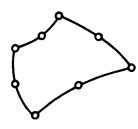
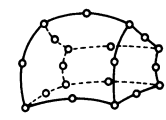
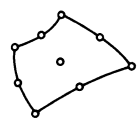
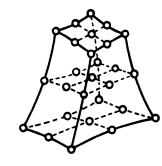
FEMGEN element types follow the convention of two characters indicating dimensionality of the element:

BE for beam elements
TR for triangular elements
QU for quadrilateral elements
PE for pentagonal or five sided elements (ie, wedges)
HE for hexagonal or six sided (ie, brick elements)

followed by the number of nodes in that element type, thus:

TR3 is a three-noded triangle
HE20 is a twenty-noded brick element

The library of elements available in FEMGEN is shown below:

eltype	element	eltype	element
BE2		PE6	
BE3			
TR3		PE15	
TR6		PE18	
QU4		HE8	
QU7		HE16	
QU8		HE20	
QU9		HE27	

1.7 THE DATA BASE

FEMGEN builds and maintains a structured file of data representing all the geometric and mesh parts which are defined directly by the user and which are generated indirectly as a result of user instructions.

During the execution of FEMGEN the data structure is held in a work file either in a random access disk file or in virtual memory accoring to the capabilities of the hardware.

On exiting from FEMGEN the user makes a choice to determine whether

a) all input during that session is discarded
b) the current data is stored in a compressed form in a sequential file and
    (i) replacing the data picked up at the beginning of that session, or
    (ii) producing an additional version of that data, ie, keeping both 'before' and 'after';

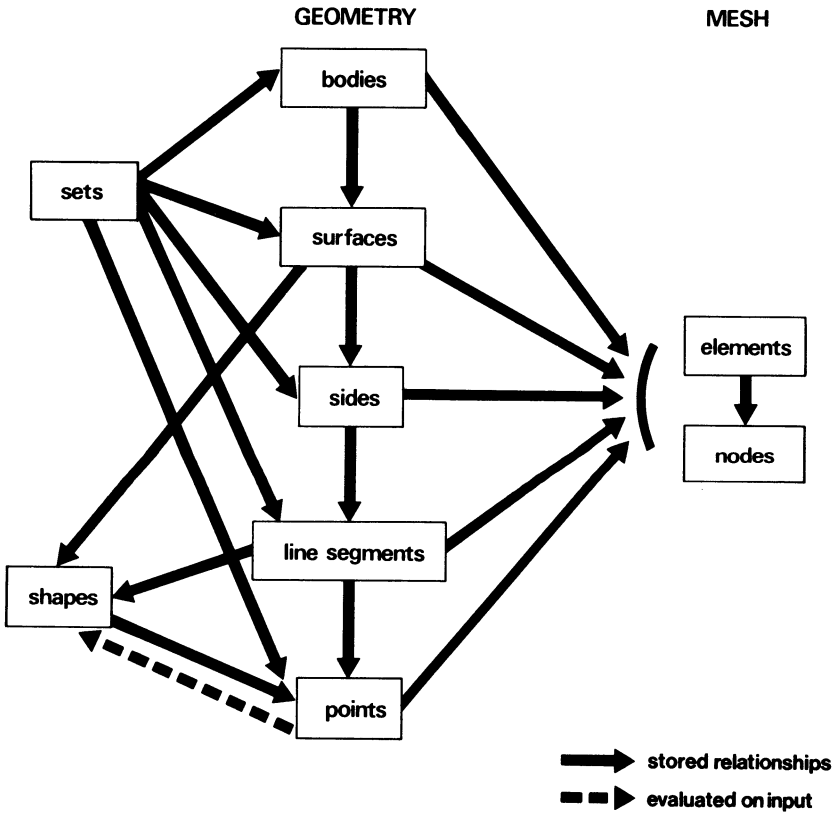
During execution the user must control when the data is saved for future use, eg, just before attempting some ambitious sequence of commands.

Other files may be produced or required during execution of FEMGEN as follows:

produced - by hardcopy plotting
         - by logging commands
         - by dumping geometry to a file formatted for
              reinput to FEMGEN or
              input to FEMVIEW
required - by a request to read commands from an offline file

Access to the FEMGEN data base is also provided outside execution of FEMGEN and under full user control via the LIBRARY OF USER ROUTINES described in 1.3 above.

The following diagram summarises the components and their relationships the FEMGEN data structure:



1.8 THE COMMAND PROCESSOR

The input to FEMGEN is given using simple and natual commands which are written in free format, and executed one at a time immediately on entry. As each command is entered, its syntax is first checked, and then its logical validity within the current operating context. If these are satisfactory, the command is executed. The user has thus immediate feedback after data entry.

Commands may be entered as text strings from the keyboard, or completed using cursor hits on the graphics screen. Additionally, co-ordinate data may be supplied using a digitizing tablet.

Keyboard commands have a simple free format consisting of an alphaneric keyword followed by descriptive parameters. Thus:

```
PNT  P1  1.2  2   defines a point;

MESH             generates a mesh; and

DISP M ALL  45 25 displays all of the mesh
                  as viewed from a given direction
```

The command name and parameters need not be repeated if they are the same from one command to the next, thus reducing the amount of input required. A HELP facility will prompt the user with valid command formats when requested.

1.9 INPUT AND OUTPUT DEVICES

Input for FEMGEN may be supplied using a terminal keyboard, a graphical cursor on the screen, a digitiser, or by reading commands from an existing disc file.

Output from FEMGEN is normally directed to the screen of a graphics terminal, but hard-copy plots can be made by sending drawing instructions to a plot file in a standard output format. Other output is sent to disc files which receive logging information, and Node and element data suitably formatted for further processing by other packages such as the receiving Finite Element Analysis program.

The main body of the FEMGEN code has been designed to be device independent with carefully constructed interfaces to tailor-made device driving routines for each installation.

1.10 OTHER FEGS PROGRAMS

FEMGEN creates input data for processing by Finite Element Analysis Programs which, in turn, expand the input considerably, creating voluminous sets of results data.

The interpretation of such large quantities of data is a time consuming and error-prone activity. FEGS markets FEMVIEW, a companion program to FEMGEN, which is designed to assist engineers in the evaluation of results from a Finite Element Analysis. FEMVIEW provides comprehensive facilites for selecting, viewing and displaying results using specialised database, and interactive graphics techniques.

The FEGS post-processor FEMVIEW

Dr John H Lamont and Mr Simon H Warde

FEGS Ltd 2 All Saints Passage Cambridge England

The FEGS post-processor FEMVIEW

1.1 INTRODUCTION

FEMVIEW is an interactive graphics post-processor for the Finite Element Method of analysis. In theory it may be interfaced to any finite element analysis program, and thus can provide the user with a common post-processing facility over the range of his finite element analysis requirements.

The use of a finite element analysis program comprises essentially three stages:-

1 preparation of the model data (pre-processing)

2 analysis of the model

3 assessment of the results (post-processing)

Whilst the development of better algorithms and advances in computing technology have enabled finite element techniques to be applied to increasingly more complex problems, the tasks of data preparation and of result assessment can be discouragingly time consuming and prone to error.

FEMVIEW addresses itself to the twofold problem of assessing that the finite element model has been generated correctly, and of enabling the user to present the results of his analysis in an informative and illuminating way.

In order to achieve this goal, a variety of techniques for model visualisation, the selection of elements, and the display of results are provided, since it is often the case that the benefit of any particular technique will vary from model to model and from view to view within the same model.

With its data base specifically designed to give analysis program independence, and its highly organised data structure giving a good interactive response, FEMVIEW can provide the finite element user with an extremely flexible and powerful graphics tool.

1.2 FEATURE SUMMARY

In displaying a finite element model FEMVIEW allows the user to compose a picture of the mesh on the basis of viewpoint and visualisation mode.

To obtain the desired viewpoint, the user may rotate the model, shift the model, or zoom in or out to highlight a particular area of interest.

The visualisation modes include a rapid outline mode for the control of rotations and the checking of connectivity of the mesh, the display of both quick (poorman's) and of full hidden line views, and the selection of faces of the model by view angle. A drawing of the full mesh or the mesh with shrunken elements is also available, while a cross-section through a 3-D model may be obtained.

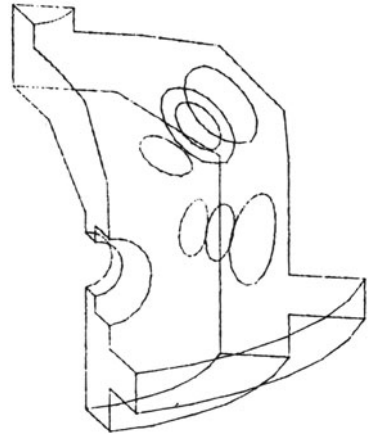
Any of the above techniques may be used on the full model or on a subset of the model which has been interactively defined. Node numbers, element numbers, groups and material numbers can be displayed, and these can form the basis for creating a subset of the model on which to present the results of analysis.

In presenting the results of analysis FEMVIEW allows the user to select the loadcase, attribute and component in which he is interested, while several presentation modes can then be used to assess these results.

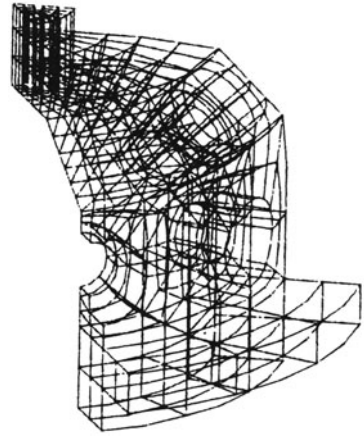
Results may be presented as numerical values, as vectors, or as contours. Nodes having result values above a certain threshold can be highlighted and also the displaced shape of a model can be displayed. Finally, different loadcases can be combined, and a graphical representation of variation for each loadcase at a particular node can be given.

Output may be directed to the graphics terminal, or to a plotter or the results may be dumped to a data file for further analysis if this is required. Specified views of a model may be saved in the bata base, and these may be accessed on subsequent occasions.

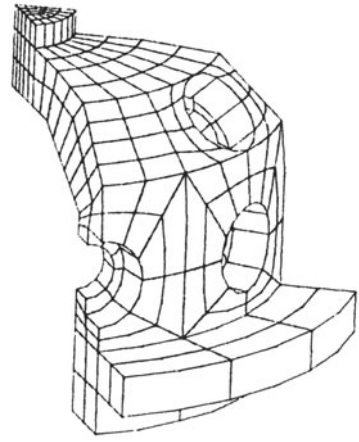
Where a colour terminal is available, FEMVIEW is designed to make maximum use of this facility in order to add information to the display of results, and to improve the user's ability to make a rapid assessment of a model.



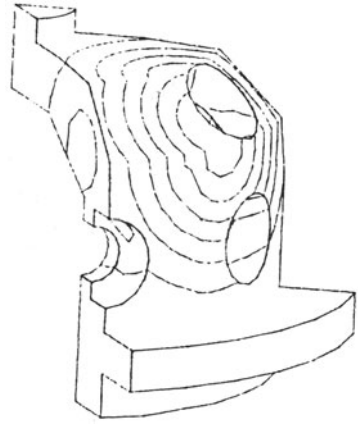
An outline view of the model is particularly useful for checking connectivity of the mesh, and for obtaining rapid model rotation. This is the default condition in which the model is represented when it is accessed from the data base.



Alternatively, when accessing the model, the finite element mesh can be drawn in full.

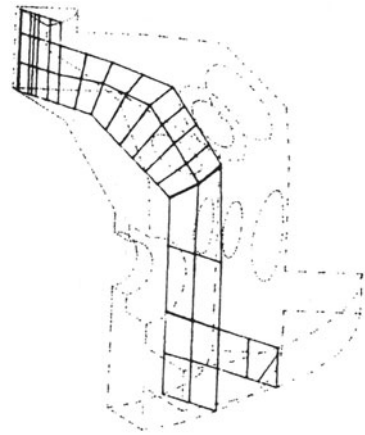


A full hidden line view provides the user with a powerful visualisation tool. Redrawing with this view mode is speeded up by retaining suface visualisation information.

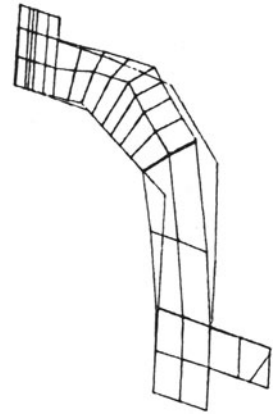


The results, displacements for example, can be contour plotted onto the surface of the model.

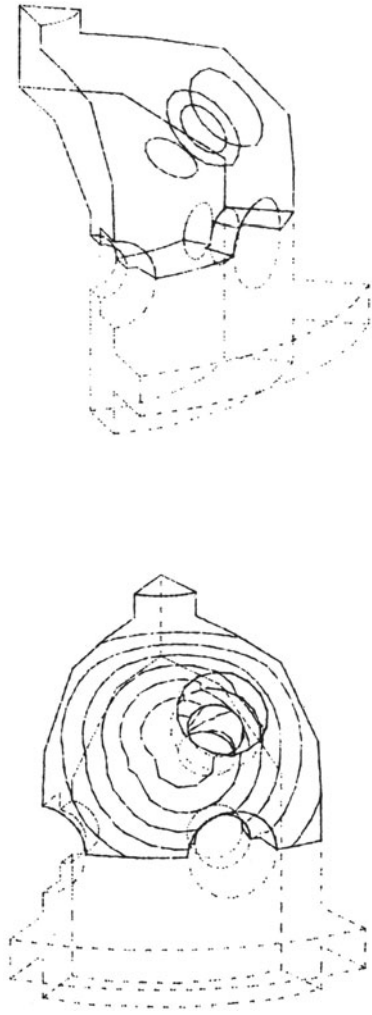
The FEGS post-processor FEMVIEW



A cross section through a 3-D model can be interactively defined.



Results can be displayed on the cross section



Results may be presented on an interactively defined subset of the model. The position of the subset may be located within the model by superimposing an edge representation of the whole model.

1.3 INTERFACING

After a Finite Element Analysis has been run, it is necessary to convert the data into the standard format specified in Appendix A. The program to perform this data conversion will vary according to the particular analysis program used.

1.4 LIMITATIONS

There is no inherent limit to the size of model which can be processed by FEMVIEW since a paging scheme is used to access information from the data base stored on disk.

At run time, however, an area of memory is reserved to hold vital information about the model, and for certain installations where a limited amount of memory is available to the user, some limitation on the model size will be encountered. A formula which allows the user to calculate the maximum size of model that can be processed is given in Appendix F. With larger computers and with virtual memory machines, these limitations will not apply.

1.5 COMPUTING REQUIREMENTS

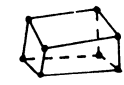
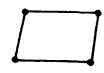
The minimum computing configuration required by FEMVIEW is as follows:-

- -32K of 16 bit words with an overlay facility
  or 128K of 16 bit words without overlay
- -1 random access disc
- -a FORTRAN compiler
- -a storage tube or raster display
- -a keyboard and cursor
- -a plotting device (if hard copy is required)

This is the minimum requirement, and FEMVIEW is designed to make effective use of systems which exceed this size to increase the performance, response, and the size of model which can be processed.

1.6 THE ELEMENT LIBRARY

The following elements can be handled by FEMVIEW. The element library covers most of the existing FEM systems libraries. A more complete description of these elements together with the convention for node numbering the elements is giving in Appendix B at the end of the manual.

eltype	element	eltype	element
BE2		TE4	
BE3		PE6	
TR3		HE8	
TR6		TE10	
QU4		PE15	
QU8		HE20	

1.7 THE DATA BASE

The Data Base is a disc file(s) holding the geometric, topological, and analysis results data required by FEMVIEW. Data is stored in the data base by the PREVIEW program and then subsequently accessed by FEMVIEW. Specific commands in FEMVIEW allow unwanted data to be released (deleted) from the data base and the storage space re-allocated.

The data base has two main parts, each stored on a separate disc file. The two parts comprise an index to the data base and the data itself. The index is designed so that finite element models may be grouped and referenced in a convenient and natural way, and has four levels :-

1 The User Index

To be able to gain access to the data base, a user must have a valid 'user-key'. This user-key can be created by the data base management routine FEMBASE which is also used to initialise the data base when FEMVIEW is first installed, or if the existing data base becomes corrupted for any reason.

2 The Project Index

Associated with a user-key can be any number of separate projects. New project codes are specified by the user when commencing a run of the PREVIEW program.

3 The Model Index

Data describing any number of models may be stored under a project code. Each model is identified by a unique name, ideally carried through from the analysis and passed to the PREVIEW program as data in the converted results file. Where the analysis program does not provide a name, one will be generated by the conversion program although this will be naturally less meaningful to the user.

4 The Load Case Index

The results of an analysis of a model are stored under its load case index. The index records the name of a loadcase and which particular results data has been stored against this model. Any number of load cases may be stored against a particular model in this way.

1.8 THE COMMAND PROCESSOR

The user controls execution of FEMVIEW by typing a command on the keyboard. A command may be one or more words long:-

```
for example 'HELP'
or 'LABEL ELEMENTS 1 TO 10'
```

where individual words in a command string are delimited by at least one space. Whenever the command processor expects input, a prompt '¾' is displayed at the top lefthand side of the screen.

The commands available to the user depend upon which module of FEMVIEW is being executed, the first words of the current commands being displayed one above the other at the right hand side of the screen. These form what is termed the current command menu. For example, when accessing a project from the data base the command menu is:-

```
DELETE  -       to delete a project
FINISH  -       to terminate run
INDEX   -       to see the index
NEXT    -       to scroll table
PLOT    -       to plot table
RETURN  -       to return to model
SELECT  -       to select project
```

To make a choice from a command menu only sufficient characters to uniquely identify the command need be typed. Thus in the above menu only the first letter of each command is required. If an incomplete command is entered the command processor will display the menu from which the next command word must be chosen. This makes it possible to construct a command without reference to a manual.

The command processor recognises the following characters as having a special meaning:

/ - kill the current command line

@ - switch the input of commands between on-line input
from the keyboard and off-line input from a data file

Selections within each menu are displayed in alphabetical order. This is to assist the user in finding a command and in deciding how many letters will have to be entered in order to uniquely identify the command.

1.9 OUTPUT DEVICES

The graphics terminal used by FEMVIEW may be a storage tube, a vector display, or a raster device, while both monochrome and colour output are provided.

Where possible the body of the FEMVIEW code is device independent, and the device dependent code is isolated within a tailor-made device driving routine which adapts the final output to the user's specific requirement.

Hard-copy is obtained by writing drawing instructions to a plot file in a standard output format. The user is then at liberty to produce a hard copy plot on the output device of his choice.

This same technique of isolating device dependent routines is used to make FEMVIEW transportable from one type of computer to another and has facilitated the implementation of FEMVIEW on a variety of machines.

1.10 OTHER FEGS PROGRAMS

FEMVIEW, although independent from the analysis program and mesh generation technique employed by any particular user, can be used to considerable advantage when associated with other programs marketed by FEGS INTERNATIONAL LTD and it's distributors.

In particular, FEMVIEW is compatible with the finite element mesh generation program FEMGEN, which in turn may be used with computer aided draughting systems and computer aided design systems like CADAM,CAM-X, CDM300, EUCLID, MEDUSA, and ROMULUS.

Appendices refer to Appendices in the FEMVIEW User Manual

A COMPUTER CODE FOR FINITE ELEMENT ANALYSIS WITH AN ENGINEERING DATA BASE.

J.R. Lévesque, H. Noé, M. Paolillo, P. Thomas

Electricité de France (Direction des Etudes et Recherches)
1, avenue du Général de Gaulle - 92141 Clamart (France)

FOREWORD

The subject of this paper is to present the basic ideas for the organization of a software package developed at ELECTRICITE DE FRANCE for static and dynamic analysis of tridimensional structures in areas such as piping, civil engineering or rotating machines. (A.V.I.S. computer code : Analysis and Vibrations of Industrial Structures).

To take into consideration the user's and the programmer's points of view, we tried to define specifications whose principal features are shortly described below :

- the simplest process for access to general purpose finite elements (FE) programs ;
- a double operating mode to build, check and maintain a FE model, either with interactive devices or by batch processing ;
- graphic and editing facilities for the display of data and results ;
- implementation of the modules for varied hardware configurations (i.e. portability on IBM 30xx, CRAY1, CII-DPSx, mega-mini, ...) ;
- reduction of efforts for maintenance and development ;
- application of virtual memory concept for the data handling in internal memory and on external supports ;
- capabilities for expanding modelling options in order to insert new finite elements or implement other algorithms.

Considering those objectives, it's now commonly admitted that one of the best solutions for modularity and flexibility is a set of independent processors communicating through a data base system.

From the F.E.M. point of view, modularity requires a versatile elements' library for substructuring methods and multilevelled superelement algorithms. For this reason, the data base concept is used to define an internal data structure, within which the FE features are described by explicit parameters. So, any processor can retrieve its data independently of the process (restart, re-analysis, post-processing, and so on...). Main features of a FE descriptor are presented.

GENERAL ORGANIZATION OF THE COMPUTER CODE A.V.I.S.

In this chapter, we describe the outlines of the computer code A.V.I.S. Applying modularity and pliability concepts led us to define a structure which may be presented as a set of independent and specific programs' modules. A data base management system provides data transfers between the different processors. The diagram of the Figure 1 shows the different steps, from the initial data files to the display of results.

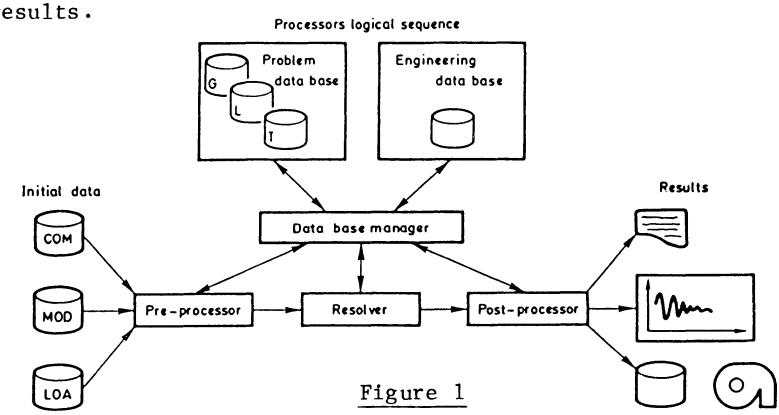


Figure 1

Initial Data

For a given problem, the user-defined data may be set off in three subsets :

- COM contains the directive statement (processors' calls, computation options, results output options, and so on, ...).
- MOD contains the data related to the FE model (materials, nodes, elements, boundary conditions).
- LOA contains applied loads data (forces, moments, ...).

In consideration of the external data grammar (not described in this paper), these three files (COM, MOD and LOA) may be separate or gathered in a same one.

Processors and Scientific Data Bases :
In the computer code A.V.I.S., a library of specialized processors is implemented. Each of them stores and retrieves its own data through a data base management system gathering a problem data base, an engineering data base and a D.B. manager.

Pre-processors are "user-oriented" modules, whose main feature is to be an entry point inside the computer code according to the analysis area (piping, civil engineering, rotating machines). They translate the initial data into internal data compatible with all the resolvers and pre-processors.

Each resolver deals with a particular analysis with a certain algorithm (static analysis, dynamic analysis by the Lanczos method, ...). Post-processors are user-oriented too. By the directive statements the user can select any option for selective editing of results. Those results may be listings, graphs or output data displayed on any device.

DATA BASE SPECIFICATIONS

In this computer code, we implemented a D.B.M.S. to achieve three objectives :
- extension of FORTRAN 77 capabilities to define objects with greather properties ;
- dynamic allocation facilities for primary and secondary storages according to the actual size of the studied problem ;
- automatic transfers of data between the workspace in core and auxiliary files.

Structure and Hierarchy of Objects :
The data base manager, called JEVEUX (= I will), recognizes three levels of objects :
- Simple object : Set of data of the same type (integer, real, character, ...) ;
- Family : Set of simple objects, built out of a reference object <RO>, whose sizes may be different ;
- Group : Set of heterogenous objects (simple objects and families) with various types.

Simple objects and families are strictly defined and assigned to a group, eventually. From a syntactic point of view, we can define the objects previously presented, in the following way :

- <RO>::=<*class*><*kind*><*type*><*dimensions*>[<*partition*><comments>] (the items into brackets denote optional arguments) ;
- <simple object>::=<name><RO>[<group name>];
- <family>::=<name><RO><number of objects>[<access mode> / <naming mode>][<group name>]

Each object is defined by its name (character string). After that, one may retrieve any object or subobject using rules of access which denote the relationship of each object :

- 'object sub-object' ;
- 'object'/'family.object'/'group.family.object' ;
- 'family'/'group.family' ; - 'group'.

The *class* defines the associated data file for further secondary storage. In the A.V.I.S. code, three classes are available : G (Global data file) ; L (Local data file) ; T (Temporary data file).

The *kind* parameter represents the nature of mathematical objects or pointers (objects assigned to one or several families) :

- E = single Element ; - RM = Rectangular Matrix ;
- V = Vector ; - SM = Square Matrix ;
- TM = Triangular Matrix (upper triangular form of a symmetric matrix).

For pointers, the *kind* parameters may be equal to :

- N = Names' pointer (character type) ;
- L = lengths of Lines' pointer (integer type) used for vectors' families or rectangular matrices ;
- C = heights of Columns' pointer (integer type) used in the same way.

The *type* parameter may take several values such as integer, real, double or character. This parameter is unfortunatly machine dependent. So it seems to be very important to take a compatible convention.

- FORTRAN 77 compilers	IBM 30xx		CRAY1
REAL	means	4 bytes (32 dig.)	1 word (64 dig.)
DOUBLE	means	8 bytes (64 dig.)	2 words (128 dig.)

- A.V.I.S. convention :
  R (Real type) 8 bytes = 64 digits = 1 word

The available types of data encountered in the computer code are :

- L = Logical, I = Integer,
- R = Real, D = Double precision,
- C = Complex, Z = Double precision complex.
- K[n] = Character*n,

The *dimensions* are defined according to the kind of the objects :

Vector ← length,
Square Matrix ← order,
Rectangular Matrix ← number of lines, number of columns.

The *partition* rules permit to access sub-object by the name and an index :

vector → sub-vector 'VEC i'
matrix → column j 'MAT i'
→ block i,j 'MAT i,j'

The *access mode* specifies characteristics of logical records on the auxiliary files :

- randomly : object by object (implicit mode) ;

- grouped : several objects gathered in the same logical record ;

- contiguously : so defined objects are stored and retrieved one behind the other without any gap.

In the same way, similar rules define the *partition* in core, and particular options intend to minimize I/O time for large objects.

Finally, *comments* may be added by the programmer to the argument list. These comments will follow the object everywhere through the self-described data base.

How to Create, Modify or Use Objects ?
The data base manager is fully FORTRAN 77 compatible. Each call to the software package JEVEUX refers to a subroutine or a function with one argument, at least : the name of the focused object. Four levels of statement are available :

- file management directive ;
- definition statements of an object with its attributes ;
- parameter setting routines ; - data assignment routines.

We describe briefly some program's calls for the three last levels.

Definition Statements. A new object or family is created using the following subroutine :

CALL JxxNOM (argument list)

for a simple object :

xx = CO (= to Create an Object)

argument list = 'name class kind type [partition key]'

for a family :

xx = CF (=to Create a Family)

argument list = 'name[.names'pointer] class kind type...
...[partition key] [access mode]', number of objects

The argument list is analyzed by the D.B. manager to insert, if unknown, the new object in the D.B. directory by "hash-coding" and create its attributes.

The other attributes, problem dependent, are initialized by separate subroutines, whose general syntax is :

CALL JCRxxx ('name', ...)

where CR means CReate. So, to define the Number Of Lines of a matrix called 'name', you write :

CALL JCRNOL ('name', nol)

or to define a regular Block PArtition into a matrix, it is :

CALL JCRBPA ('name', bl, bc)

Where bl et bc denote the numbers of lines and columns of a block respectively.

Example :
Definition of a family of square matrices with variable dimensions, random storage mode and implicit naming mode.

```
CALL JCONOM ('NORDER  class L I')
CALL JCFNOM ('MATRIX class SM R', nmat)
CALL JCRNOL ('MATRIX', JPOINT ('NORDER'))
```

The order of each item of the family is obtained through the pointer 'NORDER'.

In the same way, this family would have been declared as an hypermatrix (block partition mode), like that :

```
CALL JCFNOM ('MATRIX class SM R PB', nmat)
CALL JCRNOL ('MATRIX', JPOINT ('NORDER'))
CALL JCRBPA ('MATRIX', bl, bc)
```

Parameter-setting Routines. For high-level routines using the D.B. objects (e.g. the product of two objects), it is often desirable to get the actual values of their parameters. The data recovery is ensured by subroutines such as :

CALL Jyyxxx ('object's name', value)
yy = RV means Recover a Value,
yy = RC means Recover a Character string.

Examples :
  Number of lines of 'MATRIX' : CALL JRVNOL ('MATRIX, nol)
  Class of 'NODES' : CALL JCRCLA ('NODES', cla)

Identically, other subroutines or functions are defined :
- to create copies of an object,
- to assign initial values to the elements of an object.

Data Assignement Routines. The objects stored in the D.B. may be reached by address or by value. The D.B. manager, through the names' directory, retrieves the actual location of the object and makes sure it is present in core.

- Access by address :

The typical sequence is :

```
REAL OBJECT (1) (dummy array)
-
LOC = IWILL ('name', OBJECT)
IF (LOC.EQ.0) THEN
(error treatment)
ELSE CALL SUB (---, OBJECT  (LOC),---)
ENDIF
```

The function IWILL used as a dynamic EQUIVALENCE statement updates the relative address of the actual location of the object in central memory as an index of the dummy array OBJECT.

For families or partitionned objects appearing in a DO LOOP sequence, it's recommended to use the subroutine JAFECT for symbolic substitution of variable :

```
REAL OBJI
-
DO xx K = 1,N
      CALL JAFECT ('I', K)
      LOC = IWILL ('family.I', OBJI)
      CALL SP (---, OBJI (LOC),---)
   xx CONTINUE
```

- Access by value :

A set of functions gives the value of any element of an object for further utilization. Nevertheless, the actual capacities of the FORTRAN 77 language forbid us to assign value to the elements of an object, directly with full safety.

Values given by function xVAL ('name')

x = I(integer), R(real), C(complex), ...

Example :
JF NORDER is a heights of columns' pointer affected to a family of vectors :

```
CALL JRVNOL ('NORDER', NOL)
DO   10 I=1, NOL
     IHV = IVAL ('NORDER', I)
     DO 10 J=1, IHV
     -
10   CONTINUE
```

A better way for general computation with the D.B. objects is obtained by a library of matrix computation subroutines working on objects with compatibility tests between kinds, sizes and types parameters :

CALL PRODUC ('object 1', 'object 2', 'result')

For example, the reader will find in Appendix a subroutine which calculates the product of an object by a scalar value.

DATA BASE INTERNAL ORGANIZATION

The concept of virtual memory is introduced in the software package JEVEUX through the definition of two types of storage and the data flow management. The Figure 2 shows the diagram of internal organization of the D.B.M.S.

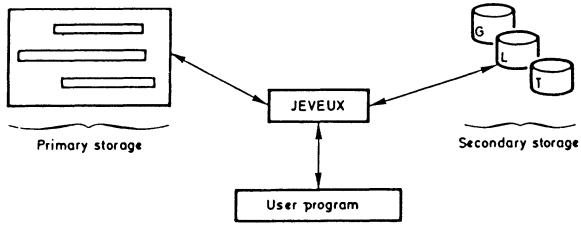


Figure 2

Primary Storage
It's a workspace in core, managed by JEVEUX, within which objects will be defined and implemented.

To let the analyst choose between two suitable program structures (sequential organization or overlay strategy), two options are available for the implementation of the primary storage.

a/ A Labelled common block with a pre-defined size before computer runs. For instance :

```
COMMON/JOBARx/IZONE (100000)
```

b/ Dynamic allocation of core not used directly following the executable load-modules, according to the value of the parameters SIZE or REGION.

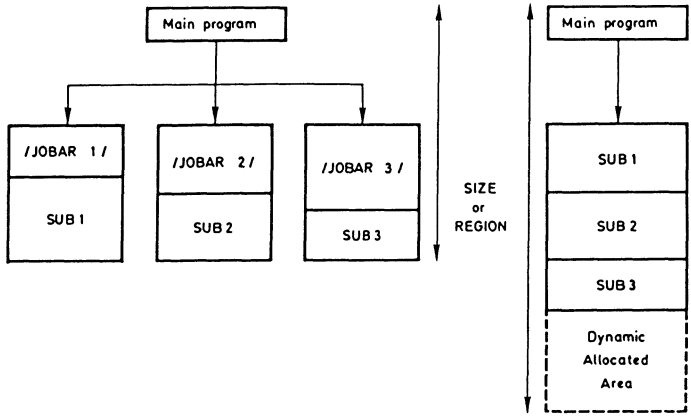


Figure 3

Though it leads to implement in the computer code several various-sized programs (static allocation) the first solution is compatible with an overlay strategy. Nevertheless, it might be interesting to implement the second solution (with dynamic allocation) for modules that need not an overlay structure. Independently of the chosen option, the internal structure is the same and is divided in two parts.

- system memory : this part is the workspace of the D.B. manager. It contains all the informations (attributes) associated to the objects and addresses which permit to point into the second part ;

- engineering data memory : that second part contains engineering data used in the program units and may be reach by the programmer with calls to JEVEUX's routines (see Figure 4).

For instance :

```
REAL TAB  (1)
-
LOC = IWILL ('object', TAB)
-
```

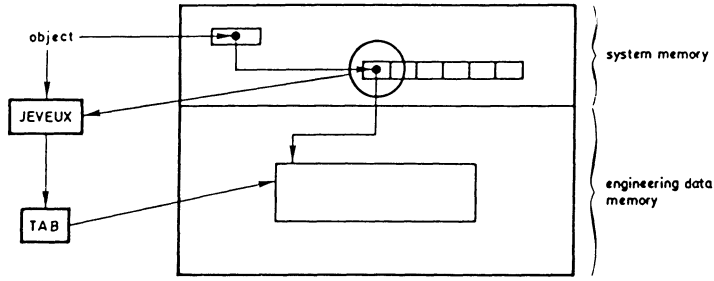


Figure 4

To increase the I/O access speeds, the primary memory is cut out of pages, each of them having a size equal to one or several K-bytes.

Secondary Storage

The secondary storage structure we defined is built out of data files with the following main features :
- direct-access data files,
- contiguous tracks (i.e. no extend).

I/O tests showed that this kind of file gives the lowest access times (IBM 30xx series).

Before storage operating, each object is cut into blocks, with respect to the associated file's blocksize (see Figure 5).

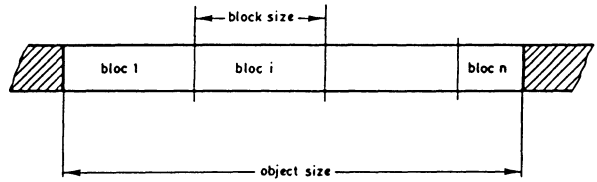


Figure 5

Retrieving the objects perform the same technique (i.e. block after block) with a particular treatment for the last block which may be uncomplete, this to avoid overlaps in the engineering data memory. If necessary, the I/O routines use buffers implemented in the system memory.

File Management

With each D.A. data file are associated two tables, kinds of "use notice" of the data file : a data file identification table and a directory completed by an attributes' table.

Data file identification table : a set of ten integer values which completely define the file and its directory's features.

- Internal variable used by I/O basic routines,
- logical unit number of the file, - block size,
- full number of blocks, - used block number,
- length (LOD) and address (AOD) of the directory,
- number of attributes (NOA) per keyword (object name),
- address of attributes' table (AAT),
- address in the system memory of the associated buffer.

Directory and Attributes' Table

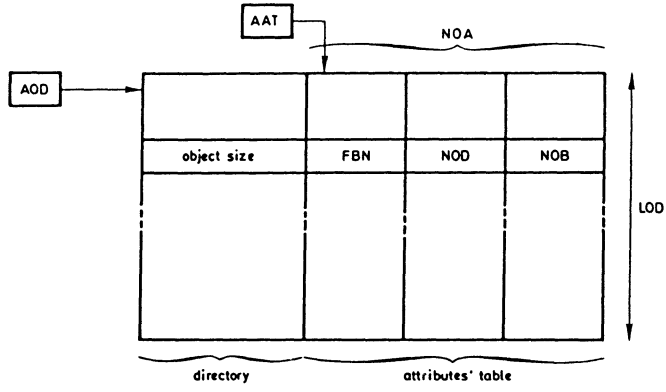


Figure 6

Every object's name is stored in the directory by the hash-coding method (the insert address is calculated using an internal coding table for each character). This highly efficient method is able to access a keyword by few tests wherever its location is in the directory.

Three attributes are associated to each keyword.

- FBN = first block number (start point of the object in the data file),

- NOD = number of data,

- NOB = number of bytes per data.

FINITE ELEMENT DESCRIPTOR

A great quantity of data characterizing a model for a given mechanical problem can be arranged in the form of element descriptors, each attached to one element of the FE model.

The data contained in the descriptor are divided into three sub-sets :

- the standard data, which depend on the nature of finite elements, and which are stored in a coded form in a catalog, implemented into the engineering data base (global data file),

- the current data, which depend on the FE model, and which are stored for each element of the model in a form determined by the catalog's directives into the engineering data base too (local data file),

- the problem data, including user's computational options corresponding to a particular mechanical application, and stored into the local data file.

The Standard Data

They give informations about the geometry and describe the set of DOF of the element. They also characterize every computational operation, specific to the type of finite element, and which have to be necessarily performed before the computer run. The standard data are divided into five blocks :

- DEScriptor Generic,
- Geometry REFerence,
- PROBlem capabilities.
- TOPOlogy,
- Degrees Of Freedom,

DESG gives numbers of items for the different blocks to dimension the generic data blocks : number of nodes, edges, faces, volumes, geometry data, DOF, ...

TOPO describes completely the element's topology to attribute nodes on edges, faces or volumes. These items are necessary for all plots (structural contour edges, hidden lines, ...), for connectivity matrix (assembly and numbering optimization algorithms).

GREF describes the reference coordinates of all nodes in the reference frame associated to the element. These undimensional coordinates, coupled with the problem data, are necessary for the definition of transformation and elementary matrices.

DOF describes the name, nature and entity type attached to each degree of freedom.

- Name : DISP (displacement) TRAN (translation),
  ROT (rotation) PRE (pressure).
- Nature : S (scalar), V (vector), T (tensor), C (torsor),
- Component ordinal number (for non scalar entities).
- Entity : Ni node i Fi face i
  Ei edge i Vi volume i
  (in relation with the block TOPO).

PROB gives indications about two kinds of operation.

- Operations which have to be performed on all the elements of the same type during a systematic step of analysis. For example, the transfer of local DOF into the global DOF system before the global stiffness matrix assembly, this for beam or plate elements.
- Operations which are available (if associated routines are implemented in the A.V.I.S. computer code), such as computation of geometric stiffness matrix or consistent mass matrix.

The Current Data

For each element of the model, they give all the informations (may be different for two elements of the same type), that don't depend on the problem. Some of them are simply red out of the user's model data, others have to be computed. The current data sub-set may be divided into five blocks dealing with systematic computation steps or standard operations on to the FE catalog.

- GEOLOC (Local Geometry) contains the actual values of elementary routines' arguments, such as nodal coordinates, lengths, areas, curvatures, ...
- FRAME defines the different frames for geometric transformations which have to be performed by the PROB block data stored in the catalog.
- TRANS gives all the parameters of these transformations.
- CONNEC (connectivity data) is a table within which one associates the location number in the global DOF system to each elementary DOF.
- MATER (material behaviour) gives the actual values of parameters which govern the material behaviour.

The Problem Data

These data are obtained by reading of the user's data, for each element. They are generally modified during the computation process. However they must remain compatible with the PROB block data, because one cannot revolve all sorts of problems with one type of element. The structure of that sub-set is the same as the current data's one, but the described operations are defined on the model by the user's directives, not by the FE catalogue's parameters.

```
      SUBROUTINE M O B S C A  ( OBJ, SCA, RES )
C
C     multiply object OBJ by scalar SCA (real) and store in object RES
C
      CHARACTER*2 TYPE, KIND, JRCGEN, JRCTYP,ITYPE
      CHARACTER*(*) OBJ, RES
      REAL TABOBJ(1), TABRES(1), SCA
      INTEGER ISCA
      DOUBLE PRECISION DSCA
      COMPLEX CSCA
C
C     is OBJ in the data-base ?
C
      IF( J E X I S T (OBJ) .NE. 0 ) THEN
C         ---> message 1
          RETURN
          ENDIF
C
C     is OBJ a simple object?
C
      IF( J R U N O B (OBJ) .NE. 1 ) THEN
C         ---> message 2 or sequence for object's familly
          RETURN
          ENDIF
C
C     is OBJ a logical's or a character's type object ?
C
      ITYPE = J R C T Y P (OBJ)
      IF( ITYPE .EQ. 'L ' .OR. ITYPE .EQ. 'K ' ) THEN
C         ---> message 3  (TYPE)
          RETURN
          ENDIF
C
C     allocation of RES object
C
      IF( J E X I S T (RES) .NE. 0 ) THEN
C
C ------------ create RES by copy of OBJ attributes
C
          IF( J C C N O M ( RES, OBJ ) .NE. 0 ) THEN
C             ---> message 4
              RETURN
              ENDIF
C
C ------------ verification of attributes for existing object RES
C
      ELSE
          IF( J U E R I F ( RES, OBJ ) .NE. 0 ) THEN
C             ---> message 5
              RETURN
              ENDIF
      ENDIF
C
C     Size determination from kind ( number of elements )
C
      KIND = J R C G E N (OBJ)
C ------------ number of lines of the object OBJ
      NL = J R U N O L (OBJ)
      IF ( KIND .EQ. 'U ' )  N = NL
      IF ( KIND .EQ. 'RM' )  N = NL * J R U N O C (OBJ)
      IF ( KIND .EQ. 'SM' )  N = NL * NL
      IF ( KIND .EQ. 'TM' )  N = NL * (NL + 1) / 2
C
C     Activation of object OBJ
C
      LOCOBJ = J E U E U X (OBJ,TABOBJ)
      IF ( LOCOBJ .EQ. 0 ) THEN
C         ---> message 6
          RETURN
          ENDIF
C
C     Activation of object RES
C
      LOCRES = J E U E U X (RES,TABRES)
      IF ( LOCRES .EQ. 0 ) THEN
C         ---> message 7
          RETURN
          ENDIF
C
C     Effective operation with xAU product of scalar A by vector U
C ( xAU is a machine dependent subroutine i.e. vectorizable on CRAY 1)
C
      IF( TYPE .EQ. 'I' ) THEN
          ISCA = SCA
          CALL I A U ( TABOBJ(LOCOBJ), N, ISCA, TABRES(LOCRES) )
          ENDIF
      IF( TYPE .EQ. 'R' ) THEN
          CALL R A U ( TABOBJ(LOCOBJ), N,  SCA, TABRES(LOCRES) )
          ENDIF
      IF( TYPE .EQ. 'D' ) THEN
          DSCA = SCA
          CALL D A U ( TABOBJ(LOCOBJ), N, DSCA, TABRES(LOCRES) )
          ENDIF
      IF( TYPE .EQ. 'C' ) THEN
          CSCA = CMPLX ( SCA , 0. )
          CALL C A U ( TABOBJ(LOCOBJ), N, CSCA, TABRES(LOCRES) )
          ENDIF
      END
```

REFERENCES :

R.H. DODSS - L.A. LOPEZ (1979)
A generalized software system for non-linear analysis.
Engineering software conference SOUTHAMPTON.

H. TAGNFORS - R.RUNESSON - N.E. WIBERG (1981)
Design and generation of program for interactive pre -fe- post processing.
New and future developments in commercial fem.

B. PRAZED - R.T. KAFTKA (1982)
Organization of PARS a structural resizing system.
Advances in Engineering software (vol. 4).

C.A. FELIPPA (1980)
A data base management in scientific computing, II data structures and program architecture.
Computers and structures (vol. 12).

J.R. LEVESQUE (1981)
Implementation of finite element methods for mechanical engineering applications.
Int. Conf. on Finite element methods SHANGHAI.

J.R. LEVESQUE - H. NOE - M. PAOLILLO (1983).
Methode de gestion des objets de programme et organisation possible pour un code de calcul de structures.
Third Int. Symposium on numerical methods for Engineering, PARIS.

A. COMTE - J.R. LEVESQUE - H. NOE - M. PAOLILLO (1983).
Essai de manipulation d'objets abstraits en FORTRAN 77.
Int. Conf. Tools, Methods and Languages for Scientific and engineering computation, PARIS.

A COMPUTER PACKAGE FOR THE OPTIMIZATION OF WELDING PROCESSES

J. C. GELIN. H. ANDRZEJEWSKI. Department of Mechanical Engineering - IUT Le Creusot - F 71200 LE CREUSOT

SUMMARY

The productivity of welding processes is mainly determined by the interaction of the various individual elements, as the joint geometry, welding amperage and voltage, the electrode shape, the stick-out and the metallurgical structure of the workpiece.

The purpose of the development package is to analyse the influence of entry parameters on the geometry and metallurgical structure of fusion lines. A digital method is used to obtain a numerical profile of numerous macrographies. Then numerical methods are applied to obtain all the geometrical specifications of the joint. A statistic module is used to draw relations and correlations between entry variables and welding parameters ; and to better select the process.

This computer package is intended to an industrial environment and computer aided manufacturing. The package has several references in mechanical and nuclear engineering applications.

INTRODUCTION

Nowadays, in industrial environments, welding systems are more and more frequently use with an automatic and actual time programming. So, the optimum choice of welding parameters in order to perform a straightfoward joint geometry is very important. Futhermore, on these automatic and microprogrammed machines, many parameters (as control of welding sequence, welding amperage, position and speed of the welding head) must be specified by the programmer, so it may be useful to have a computer package to analyse and predict processing parameters.

A scheme of the computer aided welding process is given on figure 1.

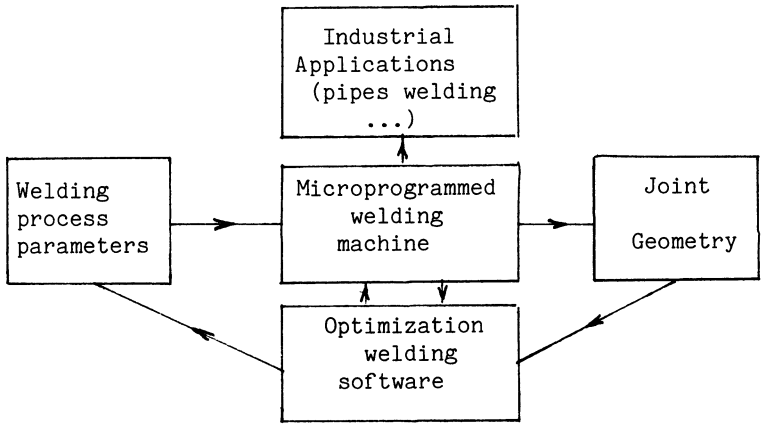


Fig. 1 - Structure of a general computer aided welding process

From our knowledge, such a software approach is the first one in welding processes. Some general studies on pulsed gas tungsten arc (PGTA) describe the influence of current frequency and shape of current welding on the joint geometry [1, 2]. For example, Becker and Adams [2] have carried out an investigation on bidimensional heat transfer in PGTA, they pointed out that the joint depth increases by using a small current temporization, at low frequencies. Siunov [3] studied the current waveform on joint metallurgy and geometry and found that the dimensions of welding joint are mainly influnced by the average value of the welding current, but that the waveform is not very significant. Slavin and al. [4] used factorial methods to analyse the joint geometry and the meaning of a large fusion lines on the structure of the material.

The software and results presented here are elaborate from an important number of welding joints. From a macrography of the joint, a digital discretisation is performed and an analytical module is used to compute the geometrical welding parameters. Different files data based on batch processing are organized. Parametric and statistics treatments of these storage files permit the identification of the parameters and the optimal choice of welding process variables.

DIGITIZATION OF MACROGRAPHIES

Theoretical aspects

The analysis of joint geometry is obtained from a sample software. This software use the following context, described on figure 2.

Let a convex domain $S \in \mathbb{R}^2$, ∂S the boundary of this domain limiting the molten area and ∂F the boundary of the test-

piece, oriented by Ox axis.

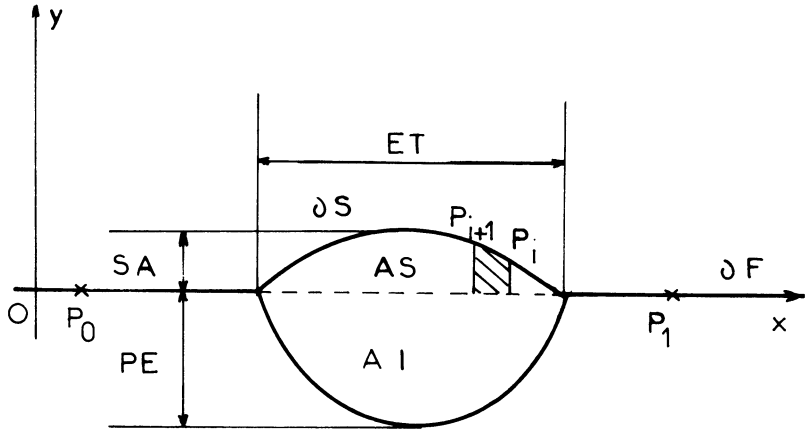


Fig. 2 - Cross section of the joint

The orientation of macrography on the digitizer tablet is obtained from the two points, P_0 and P_1. Then the joint profile is digitized in using the automatic storage information system.

Let P_i and P_{i+1} two adjacent points, in order to obtain a sufficient accuracy, these points are such that $\|P_i P_{i+1}\| < \delta$, where δ is a small pre-established quantity.

Upon completion of this procedure, a quadratic smoothing is applied for all the points such that,

$$P_i \in \{\partial S \cup \partial F - \partial S \cap \partial F\} \cdot \qquad (1)$$

So, the cross sections AS, AI and AI + AS can be easily evaluated from :

$$AS = \sum_{ij} Aij \, |\delta x_i| \, , \quad \text{for } yi \geq 0 \qquad (2)$$

$$AI = \sum_{ij} Aij \, |\delta x_i| \, , \quad \text{for } yi < 0 \qquad (3)$$

Where $\delta x_i = x_{i+1} - x_i$ and A_{ij} are the factors of the quadratic smoothing [6].

Some remarks

In welding process, the dimensions and sections on Figure 1 are

called with the following terms :

- Geometrical dilution = AI/(AS + AI)
- Joint width = ET
- Joint depth = PE
- Free joint height = SA
- Section of molten area = AS + AI = AT
- Section of inner metal = AS
- Profile joint factor = PE/ET

The process parameters are wellknown parameters in PGTA welding or arc welding.

Results of the digitizing module

Figure 3 and figure 4 show some typical joint geometries obtained from the above described module. In order to control the operation, the joint profile is recreated on a digital plotter, with an indication of welding and joint parameters.

On the other hand, the computation accuracy is evaluated from a great number of operations on the same joint, in all the cases, the maximum square is less than 2 %. Such an accuracy proves the validity of the proposed method.

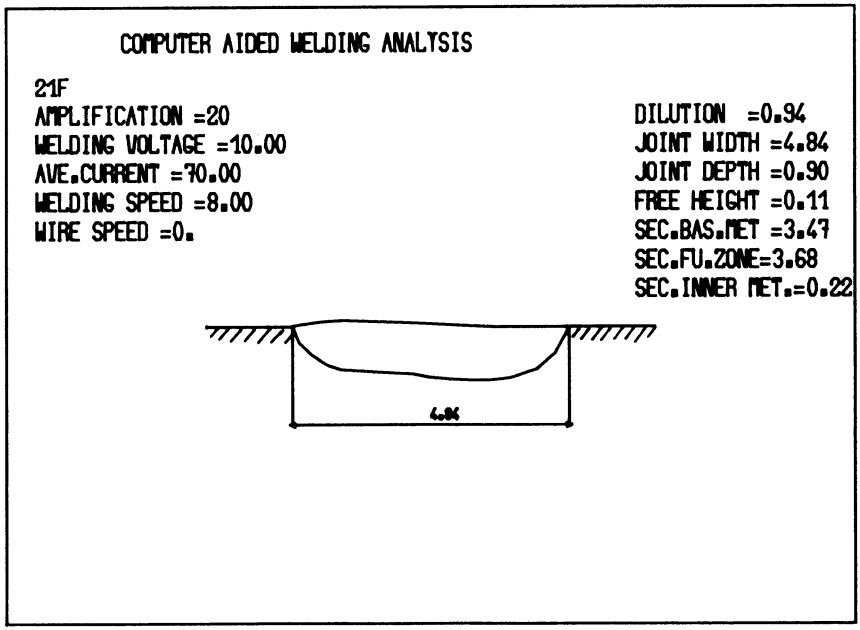


Fig. 3 - Typical joint geometry at low frequency

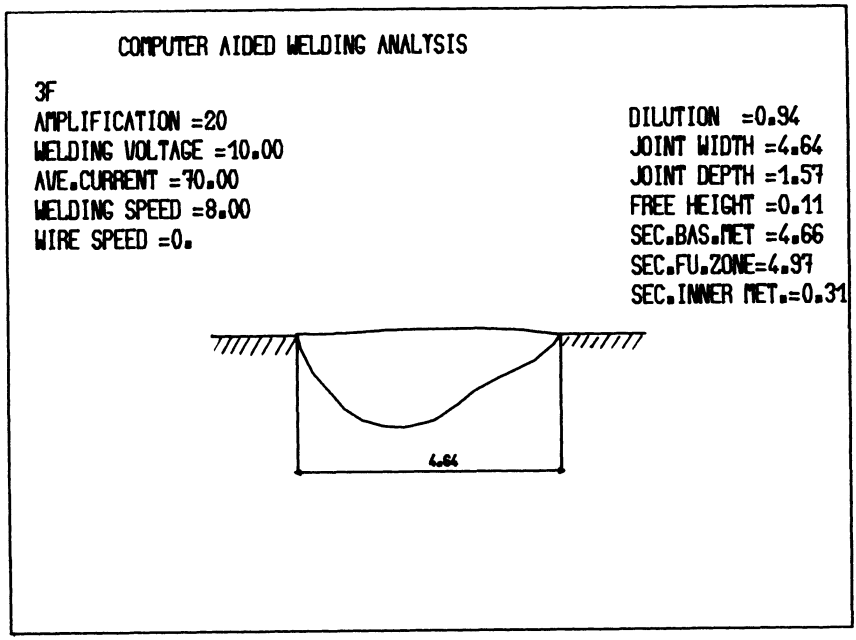


Fig. 4 - Typical joint geometry at high frequency

DESCRIPTION OF THE COMPUTER PACKAGE

The computer package "DIGIDILU" is organized around three processors of background type (Figure 5).

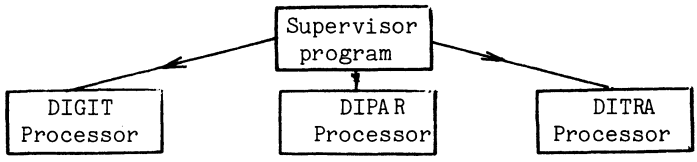


Fig. 5 - General Structure of the package

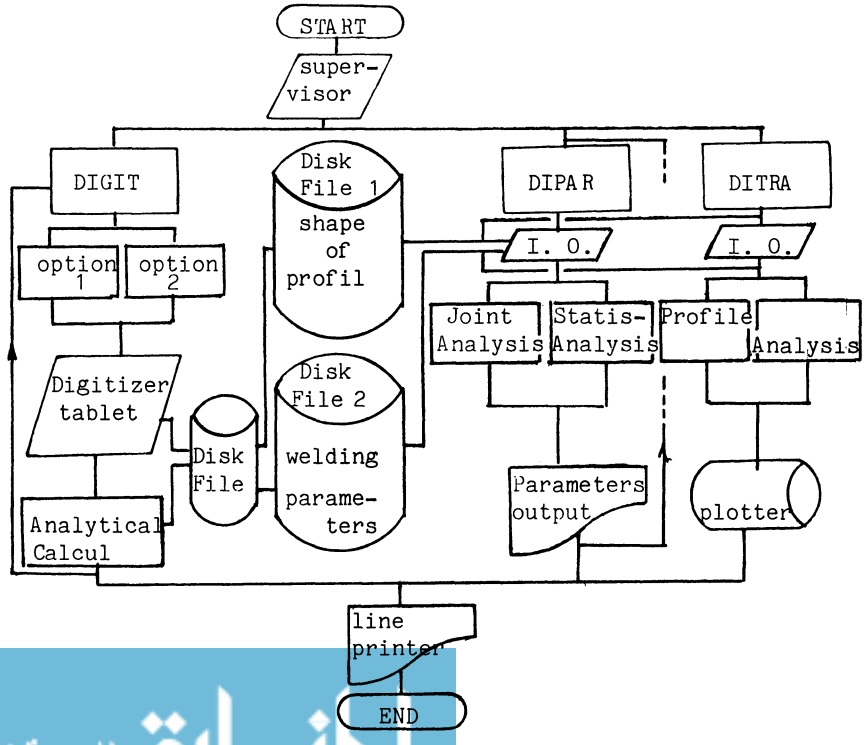
The 'DIGIT' preprocessor is an interactive set of routines to realize the data acquisition resulting from the welding joint macrography with file management routines, this processor is definitely user-oriented. The only inputs required from the the user are the welding parameters. Two different sets are avalaible ; a set of routines for arc welding, and a set of routines for PGTA welding.

The 'DIPAR' processor is a set of routines with file management system which performs selection, combination or special arran-

gement of the entry welding variables and the geometrical parameters. This processor is easy to use, flexible and efficient, with an actual dialog between the computer and the operator.

The following functions are provided by the 'DIPAR' processor.
- Automatic variation of a geometric parameter versus a process welding variable.
-Statistics functions to compute means, standard deviations, sums of cross-products of deviations, and correlation coefficients of differents observations with a great number of variables. Special functions perform a multiple linear regression analysis and multiple polynomial regression analysis for a dependant variable and a set of independant variables.

The 'DITRA' postprocessor is user oriented. Generally users expect more than the results given by a lineprinter. So the user has the possibilities to draw the joint profile on a digital plotter as on figures 3 - 4. Of course, more sophisticated results can be plotted (on a digital plotter or a graphic display terminal) :
- histograms of cumulative values,
- curves of variation of parameters,
- harmonic analysis results...
Figure 6 illustrates the general system "DIGIDILU" with the unified correspondances.



EXPERIMENTAL INVESTIGATIONS

Numerous fusion lines have been obtained on stainless stell (304 L) pipes with vertical axis, by using the PGTA process (Pulsed Gas tungsten Arc). Cross sections normal to the vertical axis and diametrically opposite to the welding starting point are taken in the pipes, to obtain macrographies of the joints (Growth . x 20).

The generator used a control module for power source, some measuring instrumentation and a standard connection for orbital heads.

The welding power source is a transistorized power source, the welding amperage being within 1,5 A to 300 A.

Up to 16 process parameters can be programmed with respect to the head welding position.

Waveform of welding current (fig. 7)

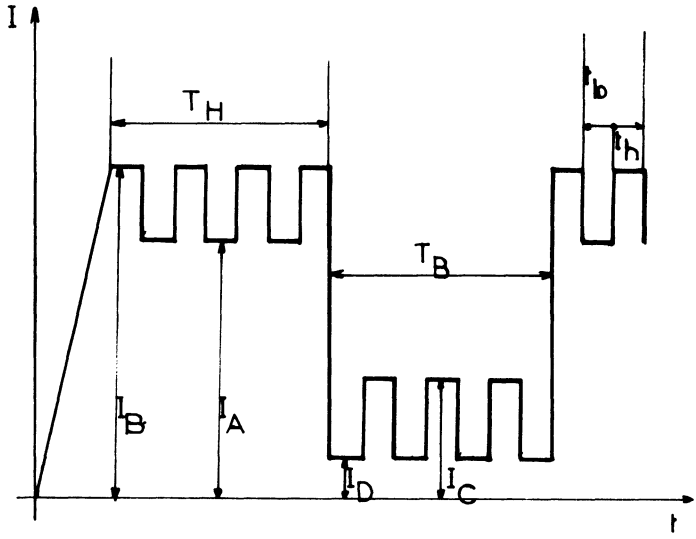


Fig. 7 - Waveform of welding amperage

First investigations

They concern the influence of pulse frequency and pulse amplitude amperage on geometrical parameters of the molten area for two power levels.

Trial conditions

* $F_p < 50\ H_z$ $\quad I_A = I_B$ $\quad I_C = I_D$ $\quad F_p = \frac{1}{T_H + T_B}$; $T_H = T_B$

* $F_p > 50\ H_z$ $\quad I_A = I_D \quad T_H , T_B$
$\quad I_B = I_C \quad t_h = t_b$

Welding speed V_s = 8 cm per min.

Two power levels are obtained with two average amperage values. I_M = 70 A, I_M = 120 A.

Pulse frequency studies

Current amplitude 60 A for average amperage 70 A.
Current amplitude 90 A for average amperage 120 A.

Pulse amplitude current studies

F_p = 400 H_z for I_M = 70 A and I_M = 120 A.

Second investigations

They concern the influence of pulse frequency for different welding speeds.
I_M = 70 A, Ap = 60 A.
V_s varies from 5 cm per min to 20 cm per min.

RESULTS

Prevailing geometric parameters are width, depth, section of molten area and shape factor.

In the first investigations the frequency 400 Hz gives a better depth, section of molten area and shape factor (Fig. 8, 9, 10, 11). We have pointed out the same results for an average amperage of 70 A and 120 A. The pulse current amplitude at 400 H_z leads to a small increase of geometrical parameters but this phenomeno is the consequence of small increase of average amperage. Therefore, current amplitude has not a significant influence on the geometrical parameters (Fig. 12 - 13).

The welding speed modifies the heat input and it was interesting to know if pulse frequency always modifies the geometrical parameters with the same variations. For a low speed (up to 14 cm per min), the geometrical parameters change with the same variations : there is a peak of shape factor for some hundred hertz (250 up to 1 000)(fig. 14 - 15). After 14 cm per min, the variations of geometrical parameters are not so distinct. However, we have two modes of variations : one at low frequency up to 50 H_z and a second one at high frequencies from 50 H_z up to 10 KH_z (fig. 16 - 17).

We do not agree with Stockinger [5] who found that shape factor is minimum for a frequency of 500 H_z, but he worked with aluminium 2 014 - T 87 and he had a generator consisting of two power sources. The current wave form is in our case different from his. However we can say, in agreement with him, that a frequency of some hundred Hertz produces high variations of geometrical parameters of the molten area.

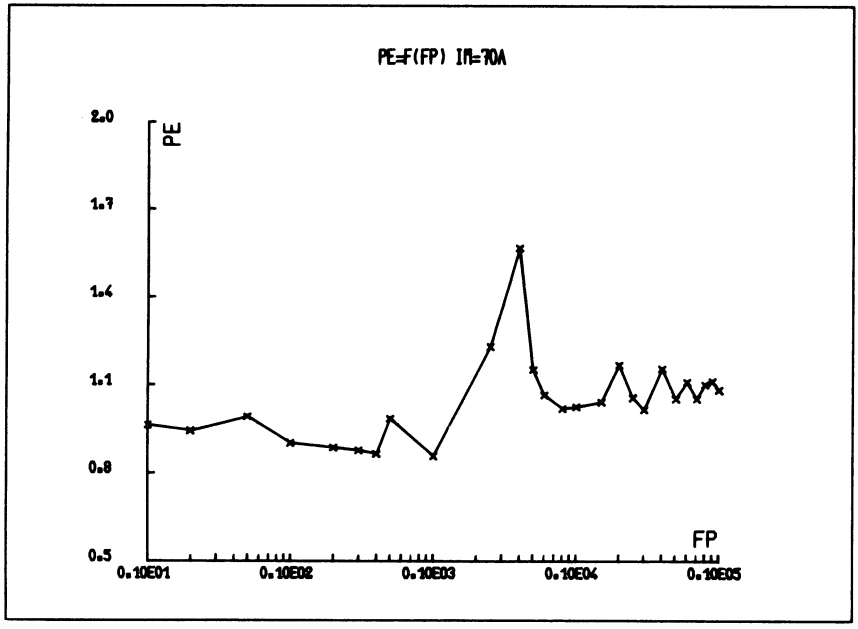


Fig. 8 - Welding depth versus pulse frequency
I_M = 70 A - V_s = 8 cm/mn

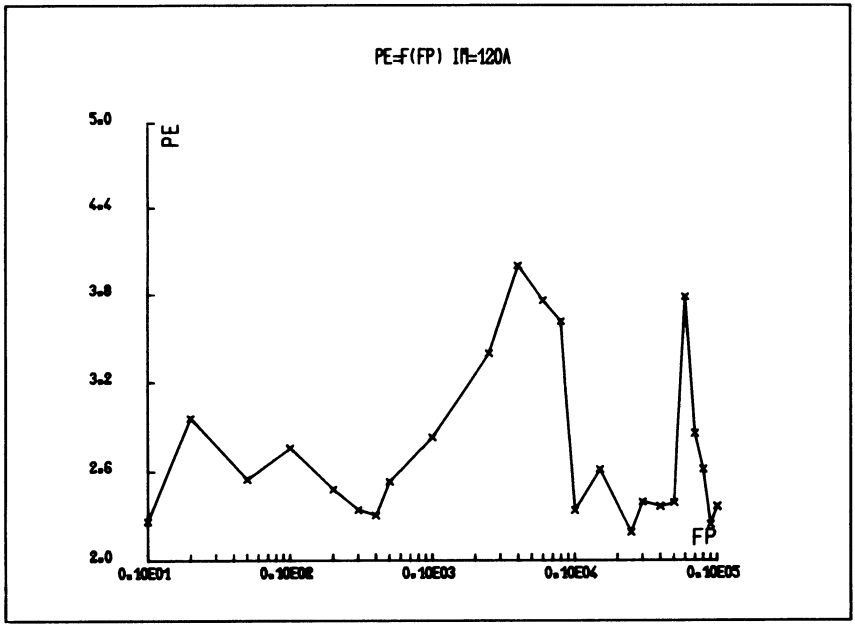


Fig. 9 - Welding width versus pulse frequency
I_M = 120 A - V_s = 8 CM/mn

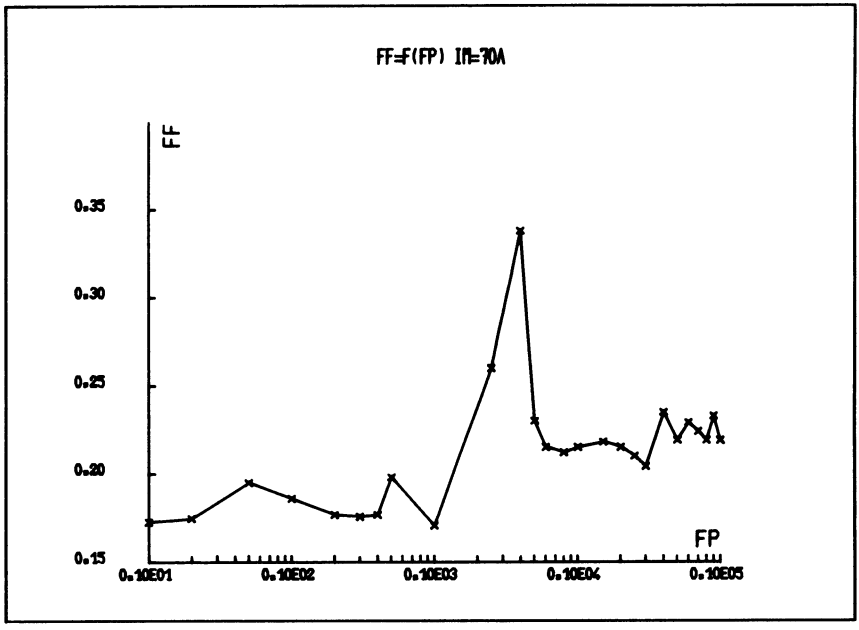


Fig. 10 - Shape factor versus pulse frequency
I_M = 70 A - V_s = 8 cm/mn

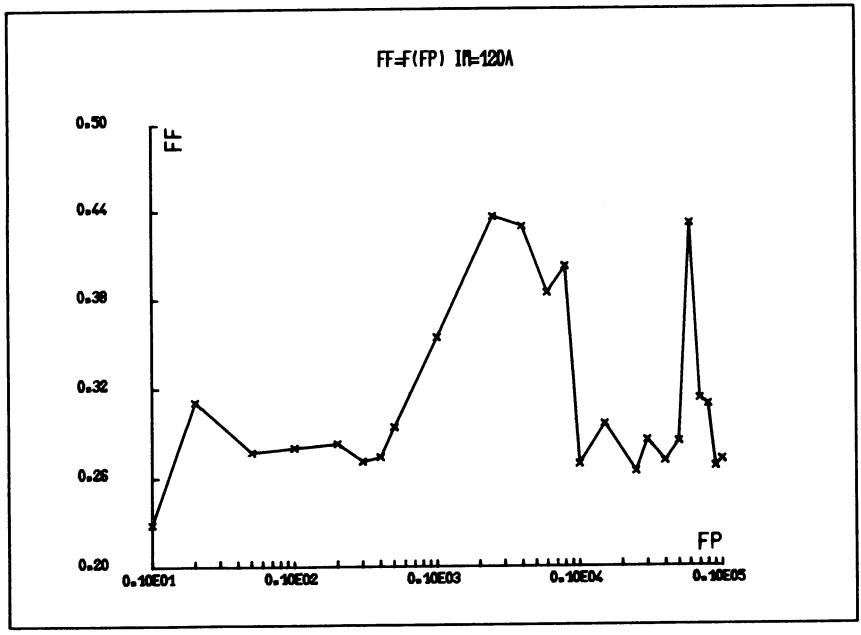


Fig. 11 - Shape factor versus pulse frequency
I_M = 120 A - V_s = 8 cm/mn

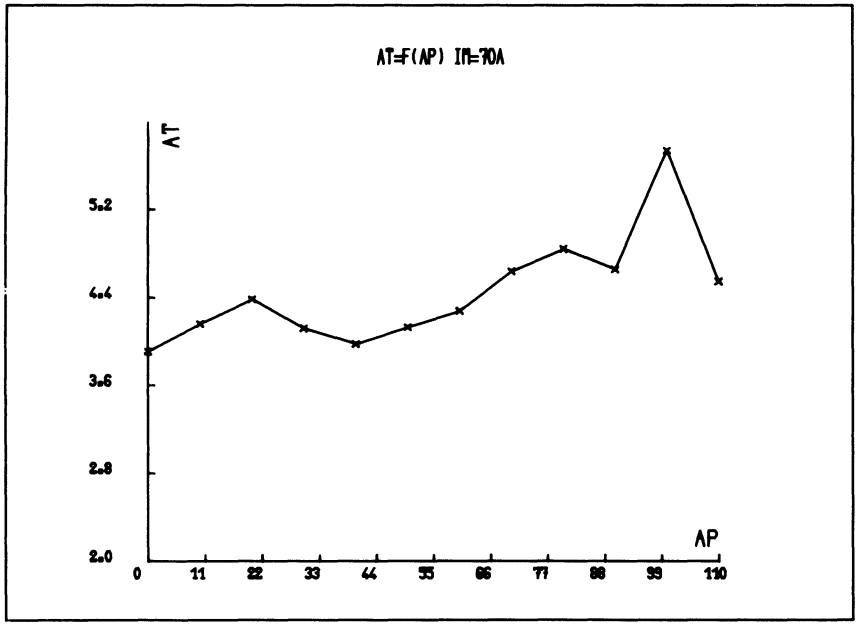


Fig. 12 - Molten area versus pulse amplitude
I_M = 70 A - V_s = 8 cm/mn

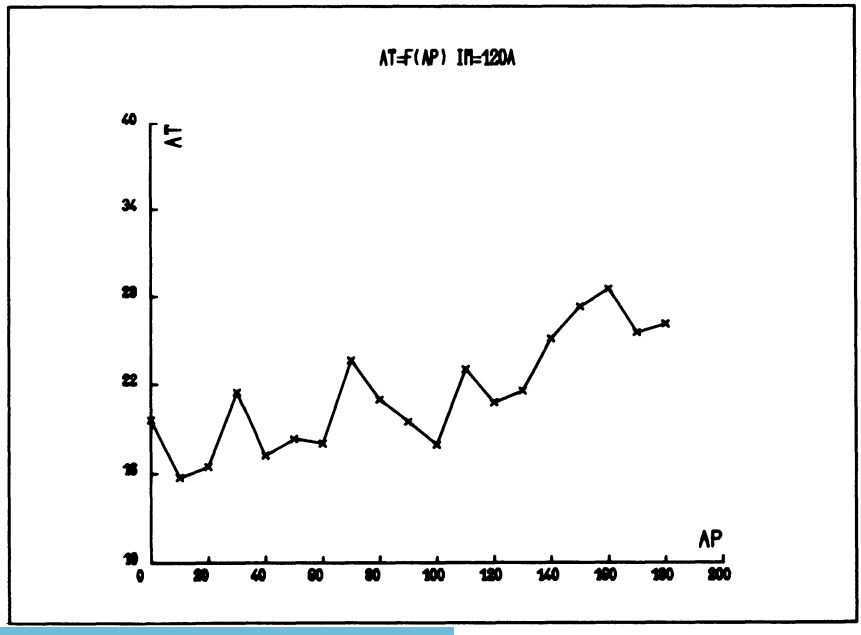


Fig. 13 - Molten area versus pulse amplitude
I_M = 120 A - V_s = 8 cm/mn

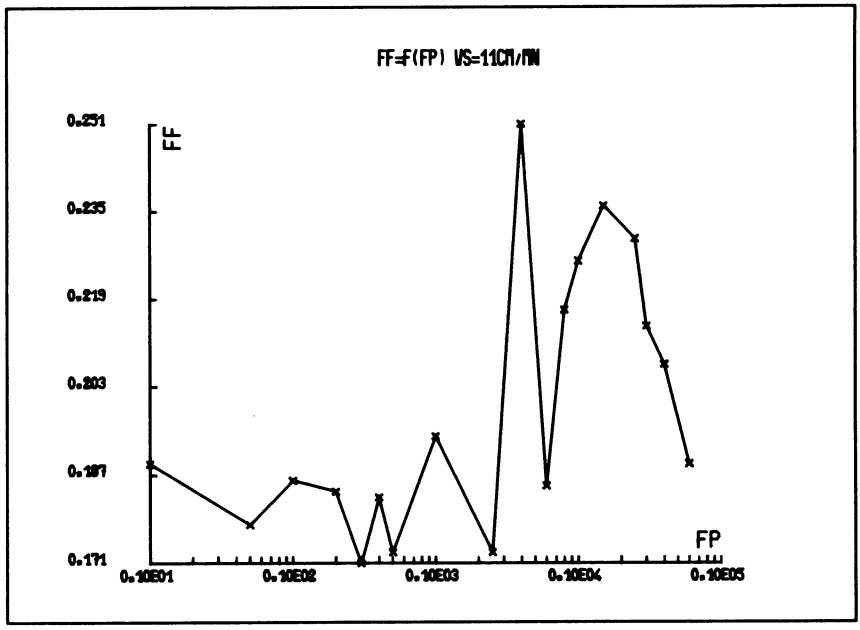


Fig. 14 - Shape factor versus pulse frequency
I_M = 70 A - V_s = 14 cm/mn

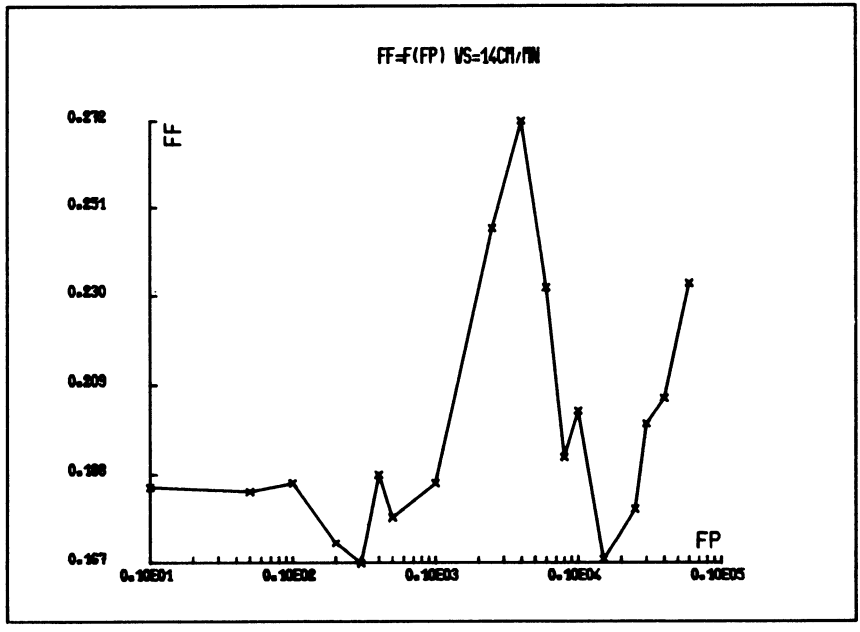


Fig. 15 - Shape factor versus pulse frequency
I_M = 70 A - V_s = 11 cm/mn

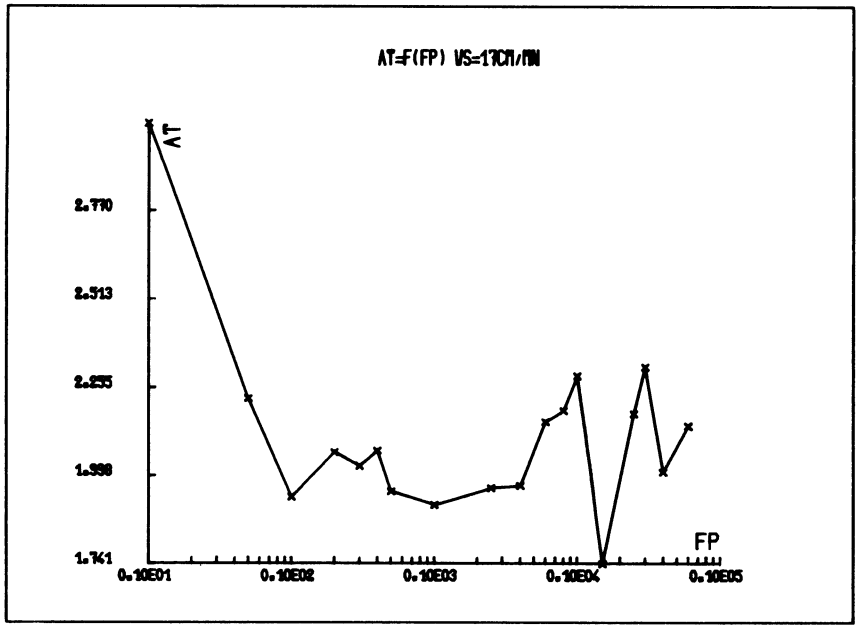


Fig. 16 - Molten area versus pulse frequency
I_M = 70 A - V_s = 17 cm/mn

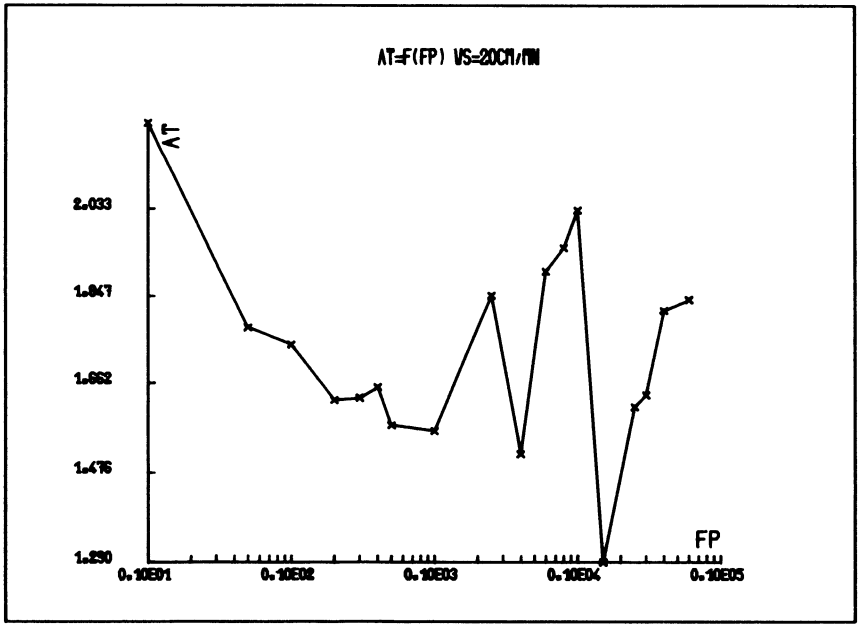


Fig. 17 - Molten area verus pulse frequency
I_M = 70 A - V_s = 20 cm/mn

CONCLUSIONS

The computer package developped is highly significant to obtain a rapid evaluation of a welding process. Digitization module is an efficient and accurate tool to determine the welding joint geometry. Parametric analysis module, the file access system and data base system permit a rapid analysis of numerous test-pieces. The statistics module can be used to establish relations and correlations between process welding variables and joint geometry parameters. For example, a subroutine compute the canonical correlations between two sets of variables, another subroutine realize a polynomial regression between joint geometry variable and a set of process variables.

From the first results, we can assert the existence of two variation shapes on geometric parameters with the frequency.

i) At low frequencies (up to 50 Hz), the geometrical parameters decrease.

ii) Over 50 Hz, there is a peak in joint depth and shape factor at frequencies within 250 to 1 000 Hz, the section parameters being at the same level than with low frequencies.

So, in industrial environments, it is necessary to operate with a frequency within 250 to 1 000 Hz, for the 304 L stainless steel. Futhermore, it would be interesting to superpose a heat pulsation with high frequency, the microprogrammed generator giving these facilities.

In order to obtain much more significant results, Fourier analysis and signal analysis are now developped to improve the optimization method of welding parameters, and to carry out a numerical simulation of welding process.

REFERENCES

[1] Rudaz A. - Synthesis rapport on the application of PGTA. Conference on Welding in the aerospace industry. Berlin - 1978.

[2] Becker D. W. and Adams C. M. - Study of Pulsed Gas Tungsten Arc parameters - Weld. Jour. - 5 - 1978.

[3] Siunov N. S. and Al. - Influence of welding amperage on the characteristics of joint in PGTA welding. Weld. Prod. 4 - 1975.

[4] Slavin G. A. and Al.- Determination from factorial analysys of the joint geometry in PGTA welding - Weld. Prod. - 9 - 1974.

[5] Stoeckinger G. R. - PGTA welding at high frequencies on Aluminium sheets - Weld. Jour. - 12 - 1973.

[6] Hacques G. - Algorithmique numérique - Armand Colin - Paris - 1971.

CALCULATION OF STEADY-STATE STRESSES IN CONTINUOUS PRESTRESSED CONCRETE BEAM STRUCTURES SUBJECTED TO CREEP AND CYCLIC TEMPERATURES

G.L. England, Y.F. Cheng, K.R.F. Andrews

King's College London, U.K.

INTRODUCTION

Prestressed concrete beams are used widely in road and motorway construction and in climates which consist of daily and seasonal variations of temperature. Their serviceable life will often be expected to exceed 100 years.

It is essential therefore that wherever possible, long-term material and structural effects should be taken into account at the design stage. Currently the Bridge Code, BS 5400, considers short-term elastic and thermal effects but does not allow for the time-dependent redistribution of stress, and its effect on structural serviceability, as brought about by the combined influence of the environmental temperatures and the inherent creep of concrete. Experimental work carried out at King's College London has revealed that significant changes to the working stresses and bending moments in prestressed continuous beams can occur during the operational lifetime of many beam structures. Not only do stresses exhibit time-dependent changes, with attendant changes being observed in bending moments also, but bending moments can at some locations change sense.

In order to assess the likely behaviour caused by short-term live load effects throughout the life of the structure it is important to have means by which the complete time-dependent behaviour under dead and/or dead plus live load can be determined. Elastic and steady-state solutions of the type described here provide these means, in the form of sensible engineering bounds to the solutions.

This paper describes a method of analysis for continuous prestressed concrete beams which is capable of evaluating section bending moments and stresses for the separate effects of mechanical loading and temperature, from elastic analyses, and the corresponding sets of quantities relating to the combined influence of creep and temperature in the long term, for both sustained and cyclically varying temperatures. The theory and a computer code 'FCREEP', or Flexibility CREEP analysis program, have been developed in such a manner that the elastic and creep solutions are obtained from the same computer routines, the different solutions corresponding simply to the different input information to the routines. This process ensures that any approximations that are adopted in the numerical formulation of the problem, influence all solutions equally. Comparisons between the various solutions of a given problem are then truly valid. The accuracy of each will be increased similarly by reducing the degree of numerical approximation inherent in the modelling process.

STATEMENT OF THE PROBLEM

Scope - The problem is that of a long prestressed continuous concrete beam on any finite number of colinear supports. Profiled prestressing tendon layouts can be accommodated and longitudinal restraints are not permitted. In other respects the problem is specified through knowledge of the applied loading and cyclical temperature states to which the structure is subjected.

It is necessary to approximate a continuous temperature variation with time over one cycle to a series of discrete temperature states which occupy the same period. The number of such temperature states is specified as part of the input data. Other information includes:

- beams are prismatic and of constant cross-section in each span; width may vary with depth.
- temperature variation over depth of section only; variations between spans.
- prestress is specified for each span; effect of cable eccentricity from centroid of section is taken into account in M_o of eq.(3).
- non-homogeneous elastic properties may be specified in each span, as a variation with depth only.
- thermal expansion behaviour is homogeneous.

- the number of temperature states for the cyclic problem must be specified; an additional datum state is also required.

- reference temperature state for creep solution must be specified.

- number of sub-divisions to the beam depth must be specified; used for I_1 to I_5 (see Table 1).

Output Data - The following information constitutes the solution:

- bending moments at the supports.

- total bending moments at selected points within each span.

- stresses at selected cross-sections for a specified number of locations in depth.

All of these quantities are output as a set corresponding to the following cases:

1. Mechanical loads only.
2. Temperature effects only, for a change of temperature from the datum to the reference state.
3. All incremental temperature changes in the temperature cycle.
4. Steady-state stresses in the reference temperature state.
5. Steady-state stresses in selected additional temperature states.

SOLUTION PROCEDURE

General - A flexibility type of analysis is adopted for the solution of the redundant support bending moments of the continuous beam.

For the elastic solution, Virtual Work equations are set up and these incorporate products of the true curvatures with successive sets of bending moments corresponding to unit values of the redundancies taken in turn.

For the creep solution Virtual Power equations (England, 1968) are constructed for the evaluation of the of the long-term limiting states of stress, viz steady-state stresses. In this part of the analysis curvature rates in the steady-state replace curvatures of the elastic solution.

For the cyclic temperature problem a steady state is defined as that state which corresponds to the existence of

repeating stresses in any part of the structure, at corresponding times in successive temperature cycles. Within any one cycle there will exist several stress states and these will differ from each other by states of thermal stresses of the same magnitude as in the thermo-elastic solution.

THEORY

Redundant Moments - For a section of the beam subjected to a bending moment, M, (hogging positive) and an axial force, P, (compressive positive) applied at the centroid of the cross-section, the following equation can be used to represent either the curvature of the section for the elastic problem or the steady-state curvature rate in the creep problem,

$$\chi = aM + bP + c \qquad (1)$$

c is an 'initial' curvature or curvature rate and there is an inherent coupling between axial and bending effects in this general representation. The parameters, a, b and c are given in Table 1; they relate to five integrals, I_1 to I_5, each taken over the depth of the beam section.

The continuity equations at the sections of the redundant moments may be written as,

$$\oint \chi \, m_i ds = 0, \qquad (2)$$

for i = 1 to N, the number of redundancies in the problem.

m_i represents the set of bending moments relating to unit value of the ith redundancy applied to the statically determinate released structure, and the integration is performed over the entire structure.

The actual bending moments are specified as the set,

$$M = M_o + \sum_{i=1}^{i=N} X_i m_i \qquad (3)$$

where M_o is the set of bending moments caused by the specified loading applied to the released structure; X_i is the ith redundancy and m_i is as before.

Substitution of Eq.(3) into Eq.(2), followed by substitution

of χ into Eq.(2) leads to a set of algebraic equations of the following form:

$$\oint \{a\ [M_o + \sum_{i=1}^{i=N} X_i m_i] + bP + c\} m_j ds = 0 \qquad (4)$$

for j = 1 to N

In matrix form the equations are:

$$[F][X] + [U] + [V] + [W] = 0 \qquad (5)$$

in which [F] is an nxn flexibility matrix, and the remaining quantities are nx1 vectors. The general terms of Eq.(5) are

$$F_{rs} = \oint a m_r m_s ds \qquad V_r = P \oint b m_r ds$$

$$U_r = \oint a M_o m_r ds \qquad W_r = \oint c m_r ds \qquad (6)$$

The solution of equation (5) reveals the redundant bending moments, [X]. In FCREEP a profile equation solution scheme is utilised to solve Eq.(5).

Internal Stresses - The section stresses, σ, (compressive positive) are evaluated from the following expression

$$\sigma = g\left\{ [1 \quad x] \begin{bmatrix} I_1 & I_2 \\ I_2 & I_3 \end{bmatrix}^{-1} \begin{bmatrix} P-I_4 \\ M-I_5 \end{bmatrix} - h \right\} \qquad (7)$$

in which h is either an 'initial' strain or strain rate (compressive positive) and is related to c of Eq.(1). g relates to the elastic or creep properties of the beam depending upon the solution being sought, and the co-ordinate x is positive when measured upwards from the centroidal axis of the beam section.

NUMERICAL EXAMPLE

The Two-Span Prestressed Beam - The beam dimensions, properties and loading are shown in Figure 1 and the section temperature distributions at particular times of a typical spring day are shown in Figure 2. It is assumed that these distributions are maintained for 6 hours each and are applied uniformly along both spans.

The steady-state stresses across the critical central support section are illustrated in Figure 3 together with

the initial thermo-elastic stresses for each temperature state of the cycle. Despite the lack of severity of slopes of the temperature distributions, the stress changes due to creep are significant, particularly in the 0800 - 1400 hours phase.

CONCLUDING REMARKS

Creep in concrete under daily or seasonal fluctations of temperature and loading has a significant adverse effect on the working stresses of prestressed concrete structures and they should be accounted for in design. A creep-temperature theory for beams is presented and a Fortran program 'FCREEP' has been developed based on the above theory and may be utilised for design or analysis. FCREEP may be run on microcomputers, minicomputers or large multiple-programming mainframes.

ACKNOWLEDGEMENTS

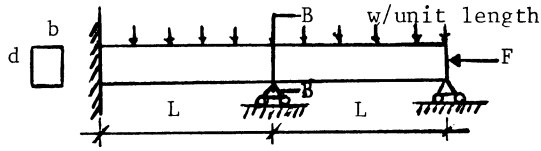
The authors wish to thank Miss Mary Emerson of the Transport and Road Research Laboratory for supplying the temperature data for the two-span beam problem.

REFERENCES

England, G.L. (1968) Time-dependent stresses in creep-elastic materials - A general method of calculation. JBSCA, Conference on recent advances in stress analysis. New concepts and their practical applications.

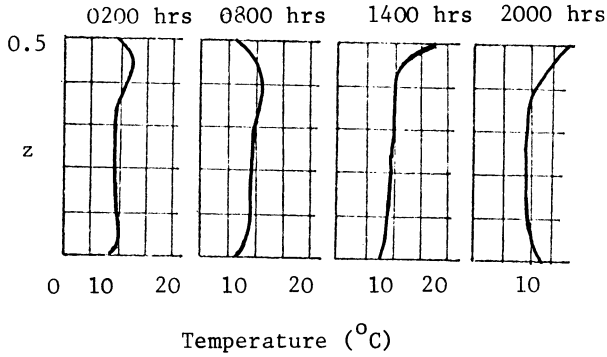
Notation	Expression
$\int_s$	Integration over the section of beam
I_1	$\int_s bg\,dx$
I_2	$\int_s bgx\,dx$
I_3	$\int_s bgx^2\,dx$
I_4	$\int_s -bgh\,dx$
I_5	$\int_s -bghx\,dx$
a	$I_1/(I_1I_3 - I_2^2)$
b	$-I_2/(I_1I_3 - I_2^2)$
c	$(I_2I_4 - I_1I_5)/(I_1I_3 - I_2^2)$

Table 1: Notations and their definitions



L = 10m, b = 0.3 m, d = 0.5 m
w = 5.10^{-3} MN/m, F = =5 MN
E = 40000 MN/m^2
$\alpha = 12.10^{-6}/^{o}C$

Figure 1: Two-span beam details



z = Distances from bottom of section

Figure 2: **Temperature distributions for two-span beam. Average spring day**

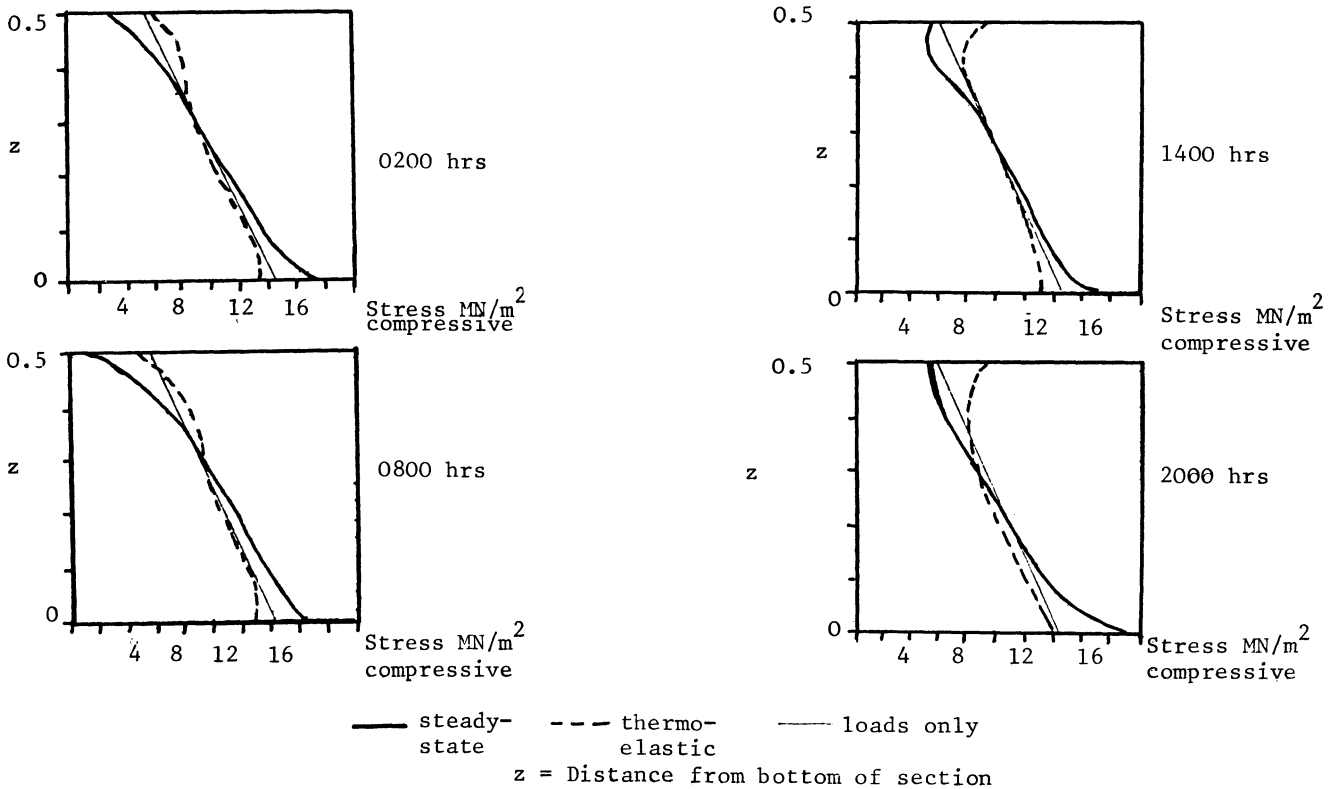


Figure 3: Stresses across section B-B of two-span beam shown in Figure 1.