A FINITE ELEMENT PROGRAM SUITABLE FOR THE HEWLETT-PACKARD SYSTEM 45 DESKTOP COMPUTER

Ray Ricardo Mallory

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A FINITE ELEMENT PROGRAM SUITABLE FOR THE
HEWLETT-PACKARD SYSTEM 45
DESKTOP COMPUTER

by

Ray Ricardo Mallory

March 1980

Thesis Advisor: G. Cantin

T191989

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Finite Element Program Suitable for the Hewlett-Packard System 45 Desktop Computer		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ray Ricardo Mallory		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 99
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Stress Analysis, Finite Element, Computer Program, Computer Graphics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The object of this work was to explore the possibility of scaling a Finite Element program to fit in a small desktop computer. A program originally written in FORTRAN IV and called STAP was selected for the present project. This program is contained in Bathe and Wilson's Numerical Methods in Finite Element Analysis and was conceived to solve static linear

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73
(Page 1)   S/N 0102-014-6601 |

(20. ABSTRACT   Continued)

elastic analysis.

 The program was rewritten in an enhanced form of BASIC language suitable for the Hewlett-Packard System 45 desktop computer. Presently, only the truss element is available; however, the program is structured so that other element types may be easily implemented. A plot of the input mesh is included in the program output.

A Finite Element Program Suitable for the
Hewlett-Packard System 45
Desktop Computer

by

Ray Ricardo Mallory
B.S., North Carolina State University, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1980

ABSTRACT

The object of this work was to explore the possibility
of scaling a Finite Element program to fit in a small
desktop computer. A program originally written in FORTRAN
IV and called STAP was selected for the present project.
This program is contained in Bathe and Wilson's Numerical
Methods in Finite Element Analysis and was conceived to solve
static linear elastic analysis.

The program was rewritten in an enhanced form of BASIC
language suitable for the Hewlett-Packard System 45 desktop
computer. Presently, only the truss element is available;
however, the program is structured so that other element
types may be easily implemented. A plot of the input mesh
is included in the program output.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

8

I.   INTRODUCTION

There is no doubt that the finite element method represents a rare breakthrough in engineering analysis. This technique, in the last twenty years, has been sharpened and refined and is today commonly utilized in stress analysis and solid mechanics. Many computer programs have been written for finite element analysis of static and dynamic problems, with most of the general purpose codes written in FORTRAN for large institutional computers. The last few years have seen the advent of the personal computer, and their memory, though limited at first, have been rapidly increasing in capacity. With today's desktop computers it now seems feasible to write a limited purpose computer program which can be implemented on a desktop computer that has enough memory capacity to handle the large memory requirements that finite element analysis requires.

This project's purpose is to write a computer program that, although simplified in various areas, shows all the important features of more general codes. The Static Structural Analysis Program (SSAP-NPS) is a simple computer program for linear elastic finite element analysis. It is written in an enhanced form of BASIC language and has been tailored to be utilized on a Hewlett-Packard System 45 desktop computer. The main objective of the program is to show that a finite element similar in

procedures and overall flow to more general codes [References 1,2] can be practical for use on a desktop computer. Only a truss element has been made available in SSAP-NPS; however, the code can be used for two- and three-dimensional truss analysis and additional elements can be added with general ease. The program also has a mesh plotting capability for verifying the input mesh.

In writing the SSAP code, the organizational structure and procedures of a FORTRAN code called STAP (Static Analysis Program), contained in Reference 3, was closely followed.

II. THE HEWLETT-PACKARD SYSTEM 45
DESKTOP COMPUTER

A. USER FAMILIARIZATION

A photograph of the H-P System 45 desktop computer is shown on Figure 1. Users of SSAP-NPS must have familiarity with its features, operation, and some knowledge of its programming language, BASIC (Beginners All-Purpose Symbolic Instruction Code). References 4, 5 and 6 are manuals which were instrumental in the creation of the SSAP-NPS program. Reference 4 contains all the information a user would need to utilize SSAP-NPS. In general, a user must be able to do the following:

1. Load and run a program stored on tape.

2. Create and size data files on tape.

3. Store and purge files on tape.

4. Create and store a program on tape.

B. FEATURES

The H-P System 45 computer for which SSAP-NPS was created has the following optional features:

1. Thermal Printer

2. Option 203 - 64K memory

3. Optional tape drive

4. ROM graphics package.

All of these features are required in using SSAP-NPS.

12

Figure 1.   The Hewlett-Packard System 45 Desktop Computer

13

The requirement for 64K memory may not be absolutely necessary but since the program itself requires about 20K bytes of memory and the only other memory option larger than 20K is 32K, it would leave only 12K of memory available for execution of the program. This would limit the program to relatively small problems.

III. <u>GENERAL FORMULATION AND IMPLEMENTATION</u>

The basic idea of the finite element method is to divide a complex problem into a series of simple inter-related problems that are easier to solve. The whole is modeled as an assemblage of discrete part, or finite elements.

In the case of a truss structure, the model is an assembly of pin-connected rods having force and displacement characteristics known from strength of materials. If each interconnecting member is identified as an element, then equations can be written expressing the element properties. The unknown forces and deflections can then be found for the overall system by combining these individual equations according to the laws of equilibrium, and then solving the resulting system of equations.

Using matrix notation, the model for a linear static system is expressed as

$$[K] \, \{U\} \;=\; \{R\}$$

where [K] is the square symmetric stiffness matrix of the assembled structure; {U} is the column matrix of unknown nodal force displacements; and {R} is the column matrix of forces applied at the nodal points.

This relationship gives the linear correlation between nodal forces and nodal displacement in the form of a set of simultaneous linear equations which is common to problems in static analysis.

After a structure has been idealized as by above relationships, the stress analysis can proceed in the following three phases:

1. Calculation of structure matrices [K] and {R}

2. Solution of equilibrium equations

3. Evaluation of element stresses

The procedure for accomplishing the above analysis phases is described only in general terms in the following discussions. For more complete discussions with illustrative examples, Reference 3 (Chapters 3, 6, 7) and Reference 7 (Chapter 24) should be consulted.

A. CALCULATION OF STRUCTURE MATRICES

The calculation of the structure matrices [K] and {R} is performed as follows:

1. Geometric, material, and loading data are read and/or generated.

2. The element stiffness matrices are calculated.

3. The structure matrices are assembled.

    1. Geometric, Material and Loading Data

Corresponding to each nodal point are a number of degrees of freedom, i.e., for the truss element there are three translational degrees of freedom at each node. An

16

array named the ID array identifies which of these degrees

of freedom are actually used in the analysis. The ID array

is of dimension number of degrees of freedom per node × NUMNP,

the total number of nodal points in the system. If ID(I,J) = 0

the degree of freedom corresponding to the Ith degree of

freedom at nodal point J is active; and if ID(I,J) = 1,

the degree of freedom is nonactive. After all active

degrees of freedom have been identified by zeros in the ID

array, they are replaced by equation numbers by columns,

starting from 1. The total number of equations is deter-

mined by the maximum value in the ID array and is called

NEQ. Nodal location with respect to a global X,Y,Z set of

axes and the loading at each node are also input at this

stage. In the program the non-zero values are stored in

an array R. Other data read or generated are the connectivity

data and material property sets. Connectivity data require

the element node numbers that correspond to the nodal point

numbers of the complete assemblage. Material property sets

identify different material properties that can be

referenced by each element.

   2.   Element Stiffness

       This phase consists of calling the appropriate

element subroutines for each element. For SSAP-NPS

only the truss element subroutine is available, but other

element subroutines could be added. Chapters 3 and

4 of Reference 3 detail the general procedure to calculate

element matrices. The nodal coordinates for the elements, material properties and loading information which have been read and stored are required here for element matrix calculations. An element matrix is stored on tape until it is needed later when assemblage to the structure matrices is performed.

For a truss element, the element stiffness matrix is the well-known matrix

$$[K_i] = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

where A and E are the element section area and E is Young's modulus obtained from the material property data input. L is the element length which is calculable from the nodal coordinates and connectivity data. The element matrix is expressed in terms of local axis. The local element matrix is transformed using direction cosines to obtain the required global element stiffness matrix for use in the assemblage of the structure stiffness matrix.

3. Assemblage of Structure Stiffness Matrix

The structure stiffness matrix [K] is calculated by direct addition of the element stiffness matrices, i.e.,

$$[K] = \sum [K_i]$$

To perform the summation, each element matrix $[K_i]$ could be written as a matrix of the same order as the stiffness matrix $[K]$, where all entries in K_i are zero, except those which correspond to an element degree of freedom. In practice, only the compacted element stiffness matrix, which is of order equal to the number of element degrees of freedom, together with a connectivity array LM are required to be stored. The connectiviy array relates to each element degree of freedom, the corresponding assemblage degree of freedom.

The distinct form of the structural stiffness matrix lends itself to two major reduction schemes that result in saving computer storage space. One is the storage of only the items within a non-zero band; and, by virtue of the matrix being symmetric, the items in the half band width are the only ones needed for the analysis. The second is to use the "skyline" technique whereby the columns above the main diagonal are stored in a one-dimensional array of length NWK. The column height information is given by array MHT. A pointer array called MAXA is used to store the addresses of the diagonal elements of $[K]$ in the A array. Chapter 6 of Reference 3 and Chapter 24 of Reference 7 give details of these two storage reduction schemes which are used in SSAP-NPS.

B. SOLUTION OF EQUILIBRIUM EQUATIONS

Program SSAP solves the simultaneous linear equations
given as

$$[K] \ \{U\} \ = \ \{R\}$$

by using a direct solution technique based on Gauss
elimination (See Reference 5). The basic procedure of the
Gauss elimination procedure is to reduce the coefficient
matrix (i.e., the items in the reduced stiffness matrix)
of the equation to an upper triangular matrix from which
the unknown displacements {U} can be calculated by a back
substitution. This process is called the Triangular
Decomposition of [K].

Subroutine COLSOL (see Section IV and Program Listing)
is used to obtain what is called the $[L][D][L^T]$ factoriza-
tion (triangular decomposition) of a stiffness matrix and
also to reduce and back-substitute the load vector. The
complete process gives the solution of the element equili-
brium equations.

C. EVALUATION OF ELEMENT STRESSES

Once the nodal point displacements {U} have been
obtained, element stresses are calculated in the final
phase of the analysis. This is accomplished by calculating
the element compacted strain-displacement transformation
matrix and then extracting the element nodal point

displacements from the total displacement vector using the LM array of the element. By establishing the strain-displacement transformations, and relating stress to strain by Hooke's law, stresses can be calculated at any desired location in the system.

IV. PROGRAM ORGANIZATION

A. SUBROUTINES

Program SSAP-NPS is written in an enhanced form of BASIC language for use on an H-P System 45 desktop computer. The program consists of a main body and 14 subroutines, each of which is designed to compute one or at most a few basic steps in the solution process. The following is a list of the subroutines and a brief description of their functions.

1.	INPUT	- Reads, generates, and prints nodal point input data to calculate equation numbers and store them in the ID array.
2.	LOADS	- Reads nodal load data, calculates load vector for each load case and stores it on file "ILOAD".
3.	ELCAL	- Loops over all element groups for reading, generating, plotting and storing the element data.
4.	ELEMNT	- Calls the appropriate element type subroutine. Only the truss element is presently available.
5.	TRUSS	- Sets up storage and calls the truss element subroutine.
6.	RUSS	- Truss element subroutine to read, generate element information, assemble structure stiffness matrix, and performs stress calculations.
7.	COLHT	- Calculates column heights.
8.	ADDRESS	- Calculates addresses of diagonal elements in banded matrix whose column heights are known.

9.	ASSEM	- Calls element subroutines for assemblage of the structure stiffness matrix. Only the truss element is presently available.
10.	ADDBAN	- Assembles upper triangular element stiffness into compacted global stiffness.
11.	COLSOL	- Solves finite element static equilibrium equations in core, using compacted storage and column reduction scheme.
12.	STRESS	- Calls the element subroutine for the calculation of stresses.
13.	WRITE	- Prints displacements.
14.	PMESH	- Plots structure mesh and numbers each node.

B. PROGRAM TAPE

The SSAP-NPS program is stored on tape in two separate files called "SSAP" and SSAPO". This tape, called the program tape hereon, is inserted in the H-P System 45 standard tape drive. In using the program, the "SSAP" program file is loaded via the keyboard into the computer memory. When the program is run, it automatically gets the "SSAPO" file when it is ready to use that portion of the program. Figure 2 shows how the program segments are organized. On the figure, the arrow indicates the part of the program to which the "SSAPO" segment is overlayed. All subroutines below the arrow on the "SSAP" segment are discarded from the computer memory since they are no longer needed.

"SSAP" Segment

```
┌─────────────────────────┐
│  ┌───────────────────┐  │
│  │       MAIN        │  │
│  ├───────────────────┤  │
│  │     ELEMNT        │  │
│  │                   │  │
│  │     TRUSS         │  │            "SSAPO" Segment
│  │                   │  │
│  │     RUSS          │  │
│  ├───────────────────┤◄─── Overlay    ┌───────────────────┐
│  │     ELCAL         │  │              │     ADDRES        │
│  │                   │  │              │                   │
│  │     COLHT         │  │              │     ASSEM         │
│  │                   │  │              │                   │
│  │     INPUT         │  │              │     ADDBAN        │
│  │                   │  │              │                   │
│  │     LOADS         │  │              │     COLSOL        │
│  │                   │  │              │                   │
│  │     PMESH         │  │              │     WRITE         │
│  └───────────────────┘  │              │                   │
└─────────────────────────┘              │     STRESS        │
                                         └───────────────────┘
```
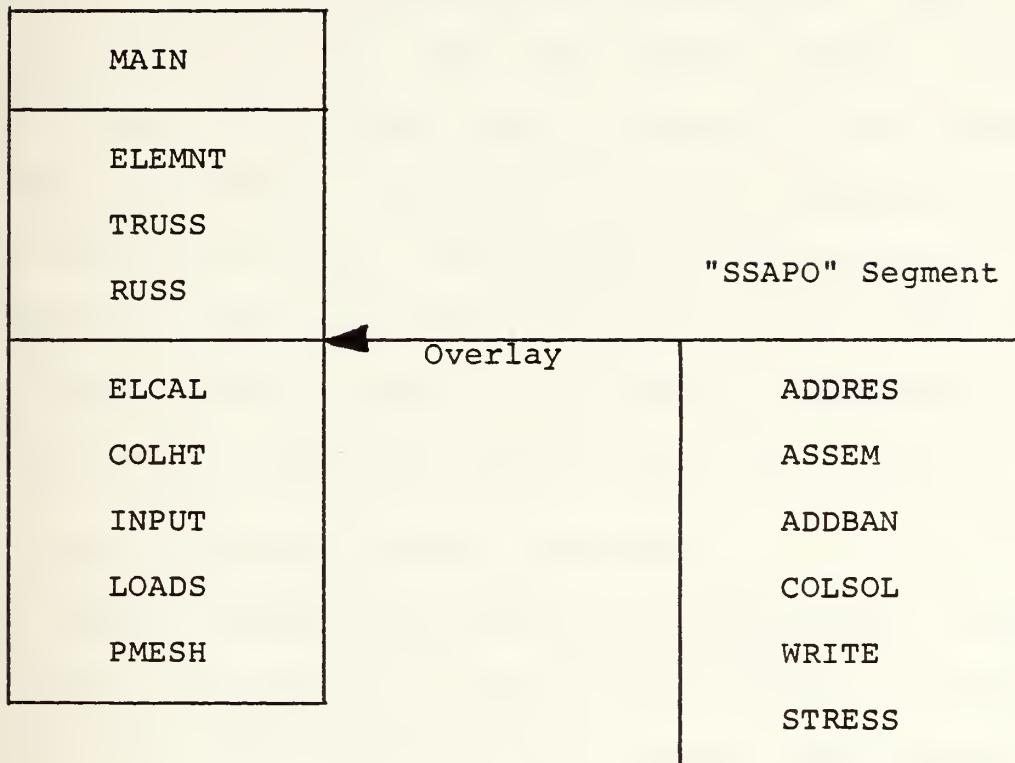
Figure 2.   Subroutine Organization And Segment Overlay

24

C. DATA TAPE

Before the program can be run, a data tape must be
inserted into the H-P System 45 optional tape drive. The
data tape contains the data files onto which computed data
will be stored and retrieved by the program. These files
are named "A1", "A2", "A3", "A4", "A5", "ILOAD", "IELMNT",
and "MESH". The "MESH" file is required by the plotting
subroutine (PMESH); the other files are discussed further
in the next section. The data tape also contains the files
which will store input data for the program. These files
are named "COOR", "LOADS", and "TRUSS". Appendix A
discusses the input data required for these files.

D. FLOW CHART AND STORAGE ALLOCATION

Figure 3 shows a flow chart of the program. Figures 4,
5, and 6 give the data files used and storage allocations
used during the various program phases. Definitions for
the variables are given in Appendix A.

```
                    ┌─────────────────────────┐
                    │ Read nodal point data   │
                    │ (coordinates, boundary  │
   START ─────────► │ conditions) and establish│
                    │ equation numbers in the │
                    │ ID array.               │
                    └─────────────────────────┘
                              │
                              ▼
                    ┌─────────────────────────┐
                    │ Calculate and store load│
                    │ vectors for all load cases.│
                    └─────────────────────────┘
                              │
                              ▼
                    ┌─────────────────────────┐
                    │ Read, generate, and store│
                    │ element data.           │
                    └─────────────────────────┘
                              │
                              ▼
                    ┌─────────────────────────┐
                    │ Plot mesh.              │
                    └─────────────────────────┘
                              │
                              ▼
                    ┌─────────────────────────┐
                    │ Read element group data, and│
                    │ assemble global structure│
                    │ stiffness matrix.       │
                    └─────────────────────────┘
                              │
                              ▼
                    ┌─────────────────────────┐
                    │ Calculate L*D*L^T factorization│
                    │ of global stiffness matrix.│
                    └─────────────────────────┘
                              │
 FOR EACH LOAD CASE           │
  ┌───────────────────────────┤
  │                           ▼
  │                 ┌─────────────────────────┐
  │                 │ Read load vector and calculate│
  │                 │ nodal point displacements.│
  │                 └─────────────────────────┘
  │                           │
  │                           ▼
  │                 ┌─────────────────────────┐
  │                 │ Read element group data and│──► END
  │                 │ calculate element stresses.│
  │                 └─────────────────────────┘
  │                           │
  └───────────────────────────┘
```

Figure 3. SSAP-NPS Flow Chart

26
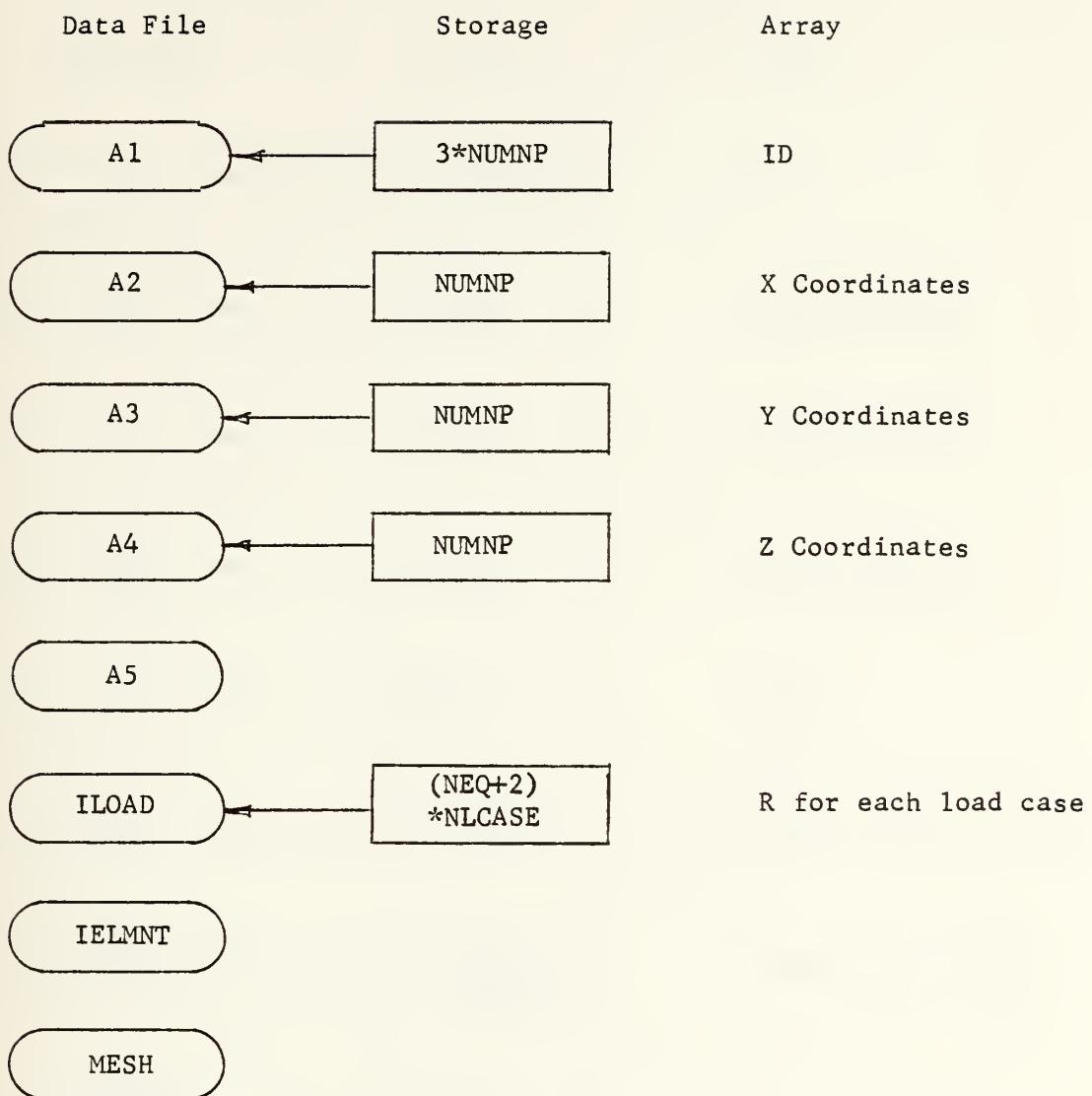
Figure 4. Array Storage Allocation and Tape Data
Location, Input Phase
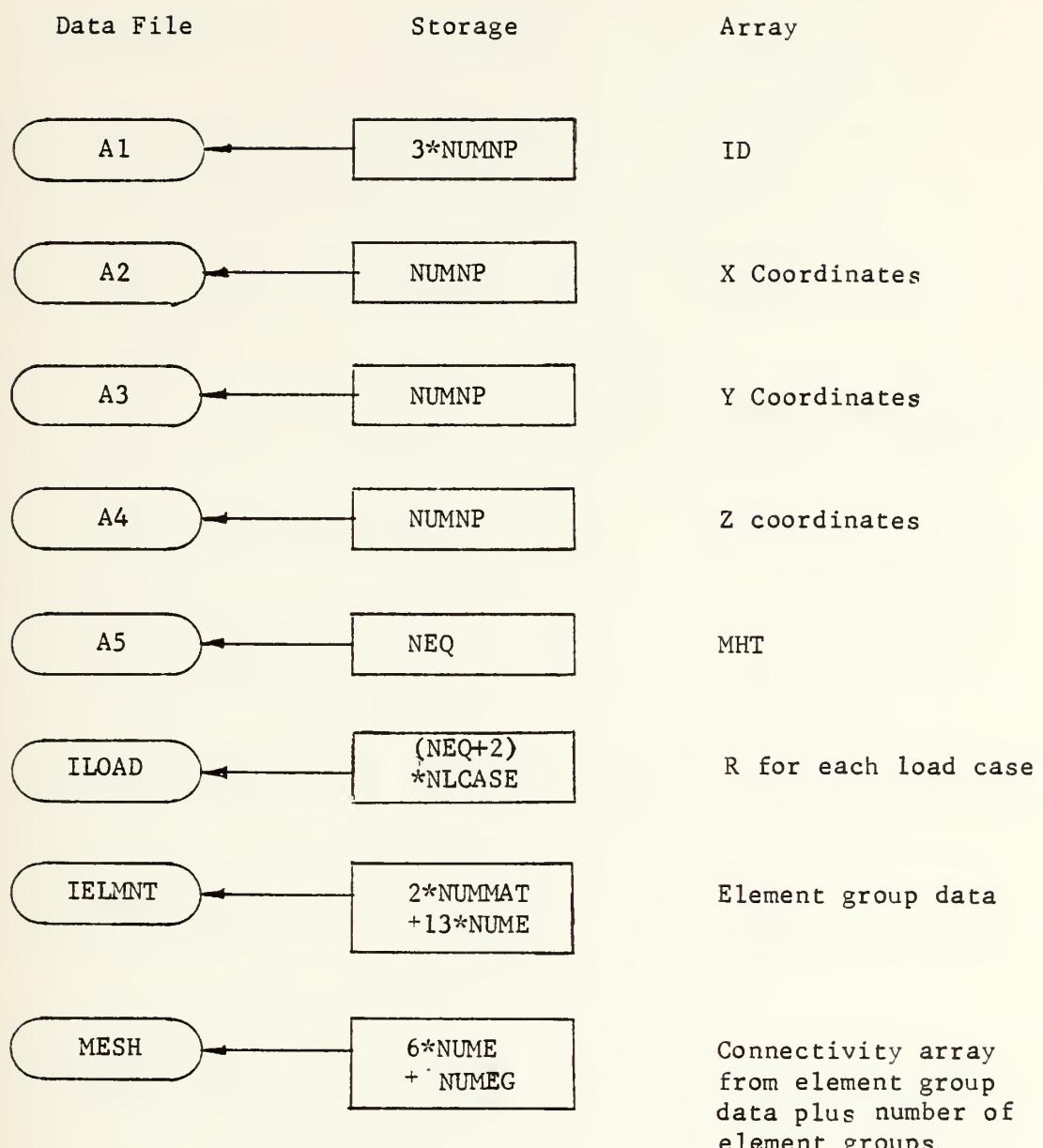
Figure 5. Array Storage Allocation and Tape Data
Location, Element Input Phase

| Data File	Storage	Array
A1	3*NUMNP	ID
A2	NEQ+1	MAXA
A3	NWK	Reduced stiffness A
A4	NEQ	U
A5	NEQ	MHT
ILOAD	(NEQ+2) *NLCASE	R for each load case
IELMNT	2*NUMMAT +13*NUME	Element group data
MESH	6*NUME +NUMEG	Connectivity array from element group data plus number of element groups
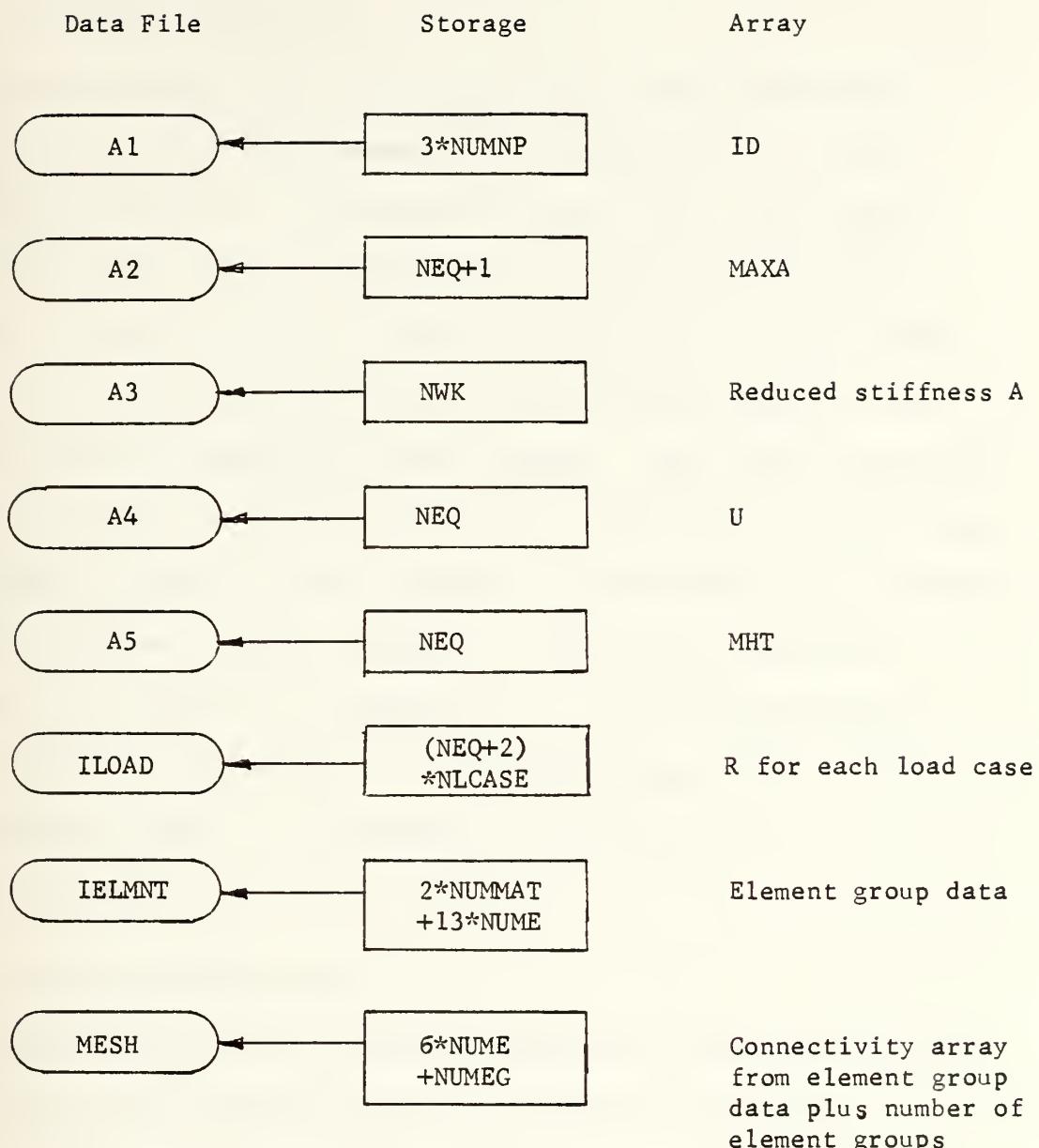
Figure 6. Array Storage Allocation and Tape Data
Location, Assemblage and Solution Phase

V. PROGRAM CAPABILITIES

A. 2-D TRUSS STRUCTURES

The SSAP-NPS program can solve for two-dimensional truss structure displacements and stresses for a given loading condition. The program output will first document the input data, automatically generating missing input if asked to do so by the input data. It will then print out a plot of the structure in the XY plane, with nodes properly labeled by node number. The nodal displacements in the X, Y, Z directions are tabulated; and finally, the axial stress for each element is tabulated. A positive stress denotes a truss element in tension; a negative stress, an element in compression. A 2-D truss example is included in Appendix B. The displacement and stress calculation phase is repeated if more than one loading case is input.

B. 3-D TRUSS STRUCTURES

SSAP-NPS can solve three-dimensional truss structures with the same capabilities as for the 2-D structures discussed above. In addition, an isometric mesh plot of the structure is outputted. An example in Appendix B illustrates a program run for a 3-D truss structure.

C. MESH PLOTS

Describing the structure model or data input seems
trivial at first but can be the source of much frustration.
Mesh points are commonly missed or tabulated incorrectly.
The computer is unforgiving and one small error can make
subsequent work meaningless. Some form of data screening
is desirable, such as a graphical display or plot of the
structure models which can be generated from a minimum of
geometric input parameters. Mesh plots will verify that
the analyst and computer are, indeed, working on the same
problem. This feature is only usable on an H-P System 45
computer that has the graphics package option installed.

SSAP-NPS produces two types of mesh plot outputs: a
mesh plot on rectangular XY axes for two-dimensional
problems where all nodes are located on the XY plane; and
an isometric mesh plot for three-dimensional truss
problems. The positive XY and Z axes for the isometric
plot are 120 degrees apart with the positive Y axis in the
vertical direction. The positive X and positive Z axes
are, respectively, 120 degrees clockwise and 120 degrees
counterclockwise from vertical. In order to avoid having
two-dimensional problems plotted on the isometric axes,
their nodes should be located on the XY plane.

VI.   REMARKS, RECOMMENDATIONS AND CONCLUSIONS

A.   RUN TIMES

In using the program, the size of the data files created
on the data tape has a significant effect on the amount of
time it takes to complete a run. This is because the pro-
gram stores and retrieves data from various locations on
the tape during a run and much time is spent winding and
rewinding the data tape.

As an illustrative case, take the 2-D truss structure
example in Appendix B. With all eleven data files
encompassing approximately 100 defined records on the
data tape, the problem took about 5 minutes from start to
finish; however, when the data files encompassed roughtly
500 defined records, the problem took more than 9 minutes
to finish. The program simply had further to go to access
the different data files during the execution of the program.
It is therefore important to size the data files such that
they are not much larger than required to hold the data
stored in them. This will require some knowledge of mass
storage techniques and definitions. One rough guide in
sizing the amount of information a data file can store is
that a standard defined record can hold 256 bytes of infor-
mation and that a real number takes up 8 bytes while an
integer number takes up 4 bytes. Therefore creating a data
file with 5 defined records means that the data file can

hold approximately $5 \times 256/8$ real numbers or $5 \times 256/4$ integer numbers or any combination thereof. For more detailed information on data storage and bytes per variable type see References 4 and 5.

B. ERROR MESSAGES

From time to time when running the program, error messages are encountered. When this happens, the program pauses. In most instances, the error can be traceable to a data input error, especially if error 17 (subscript out of range) is encountered. In cases when error 88 (read data error) or error 81 (mass storage device failure) is encountered, it may mean that the data tape has been stretched from continued use. In some cases it is curable by pressing the CONT key; otherwise the data tape may need to be re-initialized or replaced and the analysis must start from the beginning. Error 59 (physical or logical end-of-file) indicates that one or more of the data files is too small to hold all of the data to be stored in it. When this happens, the program line number at which the error occurs is displayed by the computer and the statement will reveal the data file for which the error occurred. This problem can be alleviated by enlarging the data file size, i.e., purging the existing data file and recreating it with an increased number of defined records. Other software-related error numbers and their causes are summarized in References 4 and 5.

C. RECOMMENDATIONS

As stated before, run times are greatly affected by
the winding and rewinding of the data tape. One cure for
this is to store and retrieve data from disc mass storage.
Using disc mass storage also means that larger problems
may be handled because of the increase in storage capa-
city. Alterations to the program should be minimal to affect
this change in mass storage identifier for data files.

As presently written the SSAP-NPS program can only call
upon a truss element. However, the program is structured
such that other element types can be added in the form of
additional subroutines. The addition of other element types
would also require certain subroutine modifications to allow
for the increase in the nodal degrees of freedom; and a mesh
plot subroutine would have to be written for the added
element types. The addition of other element types would
be a major improvement and would make SSAP-NPS a truly
general purpose finite element analysis program. An increase
in computer memory size would, in this case, be imperative
if the program is to be expected to handle realistically
large problems. This means the addition of some type of
disc mass storage medium and foregoing the use of the data
tape altogether.

The isometric mesh plot output is a great visual aid
for an analyst. A plot of the deformed shape of the
structure after the loads are applied is another visual aid

34

that would be very helpful in quickly assessing the effect of loads on a structure. In order to effect this for the program as presently structured, it would involve creating additional data files and rewriting the plotting subroutine. Another type of plot that would be helpful in quick analysis is the stress plot. This would indicate at each node location the magnitude and direction of the principal stress at that node so that the analyst can, at a glance, locate critical loads for a structure. The effort required for this improvement would be similar to that in effecting a deformed shape plot.

D. CONCLUSIONS

The SSAP-NPS program is a useful tool for finite element analysis of truss structure. It has a graphics capability for plotting an input mesh in 3 orthogonal planes. It also has automatic mesh generation capability.

Using a data tape for storing and retrieving data is an effective means of reducing computer memory allocation requirements; however, this comes at a price of increased run times due to tape winding and rewinding during a run. It is clear that in order to make the program more efficient that data mass storage must be on a disc storage medium.

The program is structured so as to make the addition of other element types possible by appending new element

subroutines. This is a necessary step in making the program a more general finite element program.

Additional graphics programming would also be desirable in order to fully take advantage of the H-P System 45 graphics capabilities. This could take the form of effecting a deformed structure mesh plot and a stress plot.

APPENDIX A

USER'S MANUAL

In this Appendix the data input required for the
input data files "COOR", "LOAD", and "TRUSS" on the
data tape are discussed. The units for any one problem
must be consistent. Error messages or erroneous results
in the running of the problems almost always indicate input
errors so careful preparation and review of input data is
an especially important task for the user. At the end
of the Appendix is an alphabetical list of variable names
and a brief definition of each.

A. DATA TAPE PREPARATION

Before the input data can be stored, the data tape must
have the required data files created on it, with each file's
number of defined records sufficiently large to handle the
analysis at hand. Table I is a program listing which will
create and size the required data files. The size indicated
for each data file is strictly arbitrary and were sized to
handle the example problems in Appendix B. The user can
change the record size to suit a particular analysis. This
program must be run with the data tape in the T14 mass
storage medium. If the data tape already has the data
files on the data tape but file sizes need to be altered,

Table I. Program "CREATE" Listing

```
10   ! PROGRAM TO CREATE DATA FILES FOR SSAP-HP'S PROGRAM
20   !
30   CREATE "A1:T14",2
40   CREATE "A2:T14",1
50   CREATE "A3:T14",5
60   CREATE "A4:T14",1
70   CREATE "A5:T14",1
80   CREATE "COOR:T14",5
90   CREATE "LOADS:T14",2
100  CREATE "TRUSS:T14",6
110  CREATE "ILOAD:T14",2
120  CREATE "IELMNT:T14",12
130  CREATE "MESH:T14",6
140  PRINT LIN(2),"DATA FILES FOR SSAP-HP'S CREATED ON T14.."
150  END
```

38

the program listed in Table II will purge the existing
files from the tape and the file creation program can then
be run to recreate the data files with the new file sizes.

B.   DATA FILE "COOR"

This data file contains the problem title, control
information and the nodal point information. The data
variables are recorded on the tape sequentially in the
following order:

HED$, NUMNP, NUMEG, NLCASE, MODEX

N, ID(1,N), ID(2,N), ID(3,N), X(N), Y(N), Z(N), K(N)

The second line of data variables is defined here as the
nodal information set for node N.

1.   Discussion of Data Variables

HED$ is the title information and is limited to 50
alphanumeric characters, including blanks. The title must
be enclosed in quotation marks which are not counted as
part of the 50 character limit.

NUMNP is the total number of nodes and controls the
number of times the program will look for the nodal point
information indicated on the second line of data.

NUMEG is the number of element groups. The total
number of elements are dealt with in element groups. An
element group consists of a convenient collection of
elements. Each element group is input as required by data
file "TRUSS" to be discussed later. There must be at
least one element group.

Table II. Program "PURGE" Listing

```
10  ! PROGRAM TO PURGE DATA FILES USED IN SSAP-HPS
20  !
30  PURGE "A1:T14"
40  PURGE "A2:T14"
50  PURGE "A3:T14"
60  PURGE "A4:T14"
70  PURGE "A5:T14"
80  PURGE "COOR:T14"
90  PURGE "LOADS:T14"
100 PURGE "TRUSS:T14"
110 PURGE "ILOAD:T14"
120 PURGE "IELMNT:T14"
130 PURGE "MESH:T14"
140 PRINT LIN(2),"DATA FILES FOR SSAP-HPS PURGED..."
150 END
```

40

NLCASE gives the number of loading cases to be applied to the structure.

MODEX is the flag which indicates the solution mode which gives the user the option of checking the data without executing the analysis (MODEX = 0). If MODEX = 1, the execution mode, the program goes on and solves the problem. The program gives the user the option of overriding the MODEX setting by pausing execution and asking the user if he would like to continue the solution. This is a helpful feature if the user, after reviewing the input data as listed decides that the program should go ahead and solve the problem. The user is thus saved the trouble of changing the value of MODEX and starting the program from the beginning.

N is the node (joint) number and must have a value $1 \leq N \leq$ NUMNP. Nodal data must be defined for all NUMNP nodes and may be input directly or automatically generated (see discussion of variable KN). The last node that is input must be NUMNP.

ID(1,N), ID(2,N), and ID(3,N) are, respectively, the X, Y, and Z translation boundary condition codes. A boundary condition code can only have a value of either 0 or 1. A value of 0 indicates unspecified (free) displacement; a value of 1 specifies fixed displacement, i.e., the node is not free to move in the specified direction. Concentrated forces may be applied only in the directions where ID has a value of zero.

41

One system equilibrium equation is required for
each unspecified (ID(M,N)=0) degree of freedom of the
structure. The total number of equilibrium equations is
defined as NEQ and is always less than or equal to 3 times
NUMNP. Specified (ID(M,N)=1) degrees of freedom are removed
from the final set of equilibrium equations.

$X(N)$, $Y(N)$, $Z(N)$ are the X, Y, and Z coordinates
of node N.

The nodal information sets need not be input in
node order sequence; eventually, however, all nodes from
$N = 1$ to $N = NUMNP$ must have nodal information defined.

Nodal information sets for a series of nodes

$[N_1, N_1+1*KN, N_1+2*KN, \ldots, N_2]$

may be automatically generated from information given on
two consecutive sets:

SET 1 - N_1, ID$(1,N_1)$,..., $X(N_1)$, ..., KN_1

SET 2 - N_2, ID$(1,N_2)$,..., $X(N_2)$, ..., KN_2

KN_1 is the node generation parameter given for the
first set. The first generated node is N_1+1*KN_1; the
second generated node is N_1+2*KN_2, etc. Generation
continues until node number $N_2 - KN_1$ is established. Note
that the node difference $N_2 - N_1$ must be evenly divisible
by KN_1.

42

In the automatic generation of nodal information sets, the boundary condition codes [ID(M,N)] and the coordinates [X(N), Y(N), Z(N)] for the generated sets are interpolated linearly.

C. DATA FILE "LOADS"

This data file contains the load data for the structure. Each load case requires following data entered sequentially. The total number of load cases is defined by the value of NLCASE previously recorded on data file "COOR".

LL, NLOAD

NOD, IDIRN, FLOAD

The second line of data is defined here as a nodal load data set.

LL is the load case number and must be put in ascending sequence starting with one and ending with NLCASE.

NLOAD is the number of concentrated loads applied in this load case and defines the number of nodal load data sets that will be input.

NOD is the node number to which a particular load is applied.

IDIRN is the degree of freedom number for the load. If IDIRN = 1, the load acts in the X direction. If IDIRN = 2, the load acts in the Y direction; IDIRN = 3, in the Z direction.

FLOAD is the magnitude of the load and must be acting in the global X, Y, or Z direction only.

D. DATA FILE "TRUSS"

Truss elements are two-node members allowed arbitrary orientation in the X,Y,Z coordinate system. The truss transmits axial force only, and, in general is a six degree of freedom element (i.e., three global translation components at each end). The following sequence of data is recorded on the tape data file "TRUSS" for each element. The total number of element groups (NUMEG) was defined earlier on tape data file "COOR".

NPAR(1), NPAR(2), NPAR(3)

N, E(N), AREA(N)

M, II, JJ, MTYP, KG

The second line is defined here as a material/section property data set. There are NUMMAT sets of material/ section property data sets. The data sequence on the third line is defined here as an element data set.

NPAR(1) is equal to one, the truss element type number. NPAR(2) is the number of elements in this element group. NPAR(2) is equal to the variable NUME in the program. NPAR(3) is equal to the number of material/section property data sets to be input. NPAR(3) is equal to the variable NUMMAT in the program, and must have a value equal to or greater than one. If a zero is entered, it is defaulted to one.

For the material/section property data set, N is the number of the data set and must be in ascending order beginning with one and ending with N = NUMMAT. E(N) is Young's modulus, and AREA(N) is the section area. The Young's modulus and section area of each truss element defined by each element data set is identified by one of the data sets input here.

For the element data set, M is the truss element number and must have a value $1 \leq M \leq NPAR(2)$, NUME elements must be input and/or generated (see KN discussion below) in ascending sequence beginning with one.

II is the node number at one end of the element. JJ is the node number at the other end of the element. II and JJ must not be equal and must have values between one and NUMNP.

MTYP is the value of material property data set applicable for the element.

KG is the node generation increment used to compute node numbers for missing element data sets. If KN = 0, its value is defaulted to KN = 1. If there are "J" missing element data sets between element M = K and M = L they are generated using MTYP of element number K and by incrementing the node numbers of successive elements with the value KG for element K. Table III is a summary list of variable names used in the program and a brief definition for each.

45

TABLE III - LIST OF VARIABLES

A   Reduced stiffness matrix

AREA(N)   Truss section area for material/section property
           data set N

E(N)   Young's modulus for material/section property
           data set N

FLOAD   Load magnitude

HED$ Title of problem (≤ 50 characters)

IDIRN   Degree of freedom number
               If = 1, X direction
               If = 2, Y direction
               If = 3, Z direction

ID(P,N)   Translation boundary code for direction P,
           node number N.

II   Node number at one end of truss element

JJ   Node number at other end of truss element

K   Structure stiffness matrix

K_i Element stiffness matrix

KG   Node generation increment for element data sets

KN   Node generation increment for nodal information
           sets

LL   Load case number ($1 \leq LL \leq NLCASE$)

M   Truss element number

MAXA   Array to identify diagonal element addresses

MHT   Column height array

MODEX   Flag indicating solution mode
               If = 0, data check only
               If = 1, execution

MTYPE   Material/section property set for element M

46

N	Node number ($1 \leq N \leq$ NUMNP) or material/section property data set number ($1 \leq N \leq$ NUMMAT)
NEQ	Number of equations
NLCASE	Total number of load cases
NLOAD	Total number of loads for load case LL
NOD	Node number to which load is applied
NPAR(1)	Truss element case number (=1)
NPAR(2)	Number of elements in the element group; NPAR(2) = NUME $\geq$ 1
NPAR(3)	Total number of different material/section property data set, NPAR(3) = NUMMAT $\geq$ 1
NUME	Number of elements in the element group; NUME $\geq$ 1
NUMEG	Total number of element groups; NUMEG $\geq$ 1
NUMNP	Total number of nodal points; NUMNP $\geq$ 1
R	Load vector array
U	Displacements array
X(N), Y(N), Z(N),	Coordinates of node N in global system axes

APPENDIX B

EXAMPLE PROBLEMS

In this Appendix the procedure in a typical run for the SSAP-NPS program is discussed. Then input and output for two example problems are presented.

A. TYPICAL PROGRAM RUN

In order to complete a run for the SSAP-NPS program, the user must have two tapes: a data tape, and a program tape. The data tape may or may not have the required data files already created on it. The program tape must have the following programs on it: "SSAP", "SSAPO", "CREATE" "PURGE" and "INPUT". "SSAP" is the first segment of two segments comprising the finite element program. "SSAPO" is the other segment of the program. "CREATE" is a data files creation program and is listed on Table I of Appendix A. The defined record size for each data file may be changed to suit a particular problem. "PURGE" is a tape data files purging program and is listed on Table II of Appendix A. This program clears the data tape of existing input data files if file sizes need changing. "INPUT" is a program which writes the problem input data on the three input data files as discussed in Appendix A. This program varies with each problem. Sample listings are given in the coming examples.

48

The following sequence of steps is what is involved in completing a program run. Steps 2 and 3 are omitted if the data tape already contains the data file and the user does not want to change any of the file sizes.

1. Insert the data tape in T14 mass storage, and the program tape in T15 mass storage. Turn on computer power. Execute PRINTER IS $\emptyset$ statement if Thermal printer output is desired.

2. Load and run "PURGE" program. Skip this step if the data tape does not have data files already created on it.

3. Load and run "CREATE" program.

4. Load and run "INPUT" program.

5. Load and run "SSAP" program.

The program takes over from this point. If the value of MODEX = 0, the data check only mode, the program will pause and give the user the option of either stopping the program or continuing. If the user continues the run, the program sets MODEX = 1 and completes the run. This pause in the program serves the purpose of giving the user time to review input as interpreted by the program and choosing either to go on with the calculations, or stop to make corrections and/or changes.

B. 3-D TRUSS EXAMPLE

Figure 7 shows a three-dimensional truss example with loads in the X, Y and Z directions. A tabulation of the load magnitudes acting on locations (nodes) 1, 2, 3 and 4 are included in the figure. Table IV lists the input program "INPUT" which would be required to accomplish step 4 above. Preceding comments are included here only as a means to indicate which variables are being defined by the lines that follow. Table V is the resulting output for this example.

C. 2-D TRUSS EXAMPLE

The truss in this example has two load cases as indicated in Figure 8. Table VI is the "INPUT" program required for step 4 as discussed earlier. This input program illustrates how automatic mesh generation can be utilized to minimize input data. Preceding comment statements are included to indicate which variables are being defined by the lines that follow. Table VII is the resulting output for this example problem.

Node locations for 2-D trusses should be located on the XY plane in order to avoid getting an isometric mesh plot in the output.
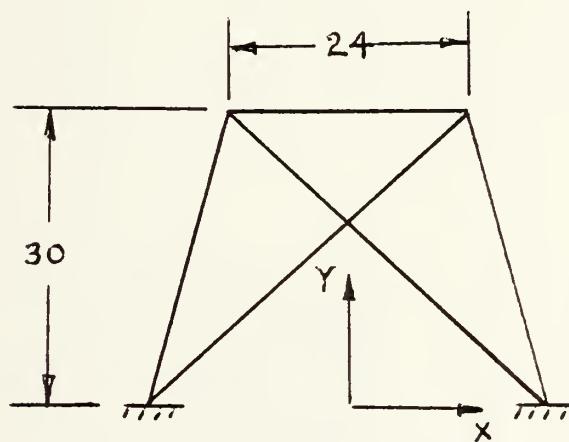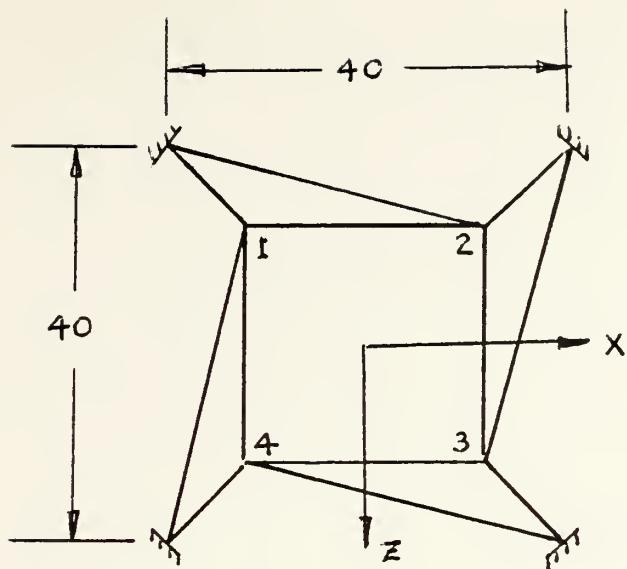
$F_{1X} = -10$ $F_{2X} = -10$ $F_{3X} = 10$ $F_{4X} = 10$

$F_{1Y} = -120$ $F_{2Y} = -100$ $F_{3Y} = -100$ $F_{4Y} = -120$

$F_{1Z} = -105$ $F_{2Z} = -105$ $F_{3Z} = -165$ $F_{4Z} = -165$

Figure 7.   3-D Truss Example

Table IV. Program "INPUT" Listing for 3-D Truss Example

```
10   ! TITLE, CONTROL, BOUNDARY, AND NODE COORDINATES DATA
20   !
30   ASSIGN #1 TO "COOR:T14"
40   READ #1,1
50   ! HED$,NUMNP,NUMEG,NLCASE,MODEX
60   PRINT #1;"SMIS TRUSS",8,1,1,1
70   ! N,ID(1,N),ID(2,N),ID(3,N),X(N),Y(N),Z(N),KN
80   ! FOR N=1 THROUGH 8
90   PRINT #1;1,0,0,-12,30,-12,0
100  PRINT #1;2,0,0,12,30,-12,0
110  PRINT #1;3,0,0,12,30,12,0
120  PRINT #1;4,0,0,-12,30,12,0
130  PRINT #1;5,1,1,-20,0,-20,0
140  PRINT #1;6,1,1,20,0,-20,0
150  PRINT #1;7,1,1,20,0,20,0
160  PRINT #1;8,1,1,-20,0,20,0
170  !
180  ! LOADING AT NODES
190  !
200  ASSIGN #1 TO "LOADS:T14"
210  READ #1,1
220  ! LL,NLOAD
230  PRINT #1;1,12
240  ! NOD,IDIRN,FLOAD AT 4 NODES
250  PRINT #1;1,1,-10
260  PRINT #1;1,2,-120
270  PRINT #1;1,3,-165
280  PRINT #1;2,1,-10
290  PRINT #1;2,2,-100
```

```
300 PRINT #1;2,3,-105
310 PRINT #1;3,1,10
320 PRINT #1;3,2,-100
330 PRINT #1;3,3,-165
340 PRINT #1;4,1,10
350 PRINT #1;4,2,-120
360 PRINT #1;4,3,-165
370 !
380 ! MATERIAL, ELEMENT CONNECTIVITY FOR ELEMENT GROUPS
390 !
400 ASSIGN #1 TO "TRUSS:T14"
410 READ #1,1
420 ! NPAR(1),NPAR(2),NPAR(3)
430 PRINT #1;1,12,1
440 ! N,X(N),AREA(N)
450 PRINT #1;1,1.0E7,.28
460 ! M,II,JJ,MTYPE,KG
470 ! FOR M=1 THROUGH 12
480 PRINT #1;1,1,2,1,0
490 PRINT #1;2,2,3,1,0
500 PRINT #1;3,3,4,1,0
510 PRINT #1;4,4,1,1,0
520 PRINT #1;5,1,0,1,0
530 PRINT #1;6,2,5,1,0
540 PRINT #1;7,3,6,1,0
550 PRINT #1;8,4,7,1,0
560 PRINT #1;9,8,4,1,0
570 PRINT #1;10,5,1,1,0
580 PRINT #1;11,6,2,1,0
590 PRINT #1;12,7,3,1,0
600 END
```

Table V. Data Output for 3-D Truss Example

SMIS TRUSS

C O N T R O L   I N F O R M A T I O N

    NUMBER OF NODAL POINTS   (NUMNP) = 8
    NUMBER OF ELEMENT GROUPS   (NUMEG) = 1
    NUMBER OF LOAD CASES   (NLCASE) = 1
    SOLUTION MODE   (MODEX) = 1
        EQ. 0, DATA CHECK
        EQ. 1, EXECUTION

N O D A L   P O I N T   D A T A

INPUT NODAL DATA

NODE NUMBER	BOUNDARY CONDITION CODES			NODAL POINT COORDINATES			MESH GENERATING CODE
	X	Y	Z	X	Y	Z	KN
1	0	0	0	-12.0000	30.0000	-12.0000	0
2	0	0	0	12.0000	30.0000	-12.0000	0
3	0	0	0	12.0000	30.0000	12.0000	0
4	0	0	0	-12.0000	30.0000	12.0000	0
5	1	1	1	-20.0000	0.0000	-20.0000	0
6	1	1	1	20.0000	0.0000	-20.0000	0
7	1	1	1	20.0000	0.0000	20.0000	0
8	1	1	1	-20.0000	0.0000	20.0000	0

GENERATED NODAL DATA

NODE NUMBER	BOUNDARY CONDITION CODES			NODAL POINT COORDINATES			MESH GENERATING CODE
	X	Y	Z	X	Y	Z	KN
1	0	0	0	-12.0000	30.0000	-12.0000	0
2	0	0	0	12.0000	30.0000	-12.0000	0
3	0	0	0	12.0000	30.0000	12.0000	0
4	0	0	0	-12.0000	30.0000	12.0000	0
5	1	1	1	-20.0000	0.0000	-20.0000	0
6	1	1	1	20.0000	0.0000	-20.0000	0
7	1	1	1	20.0000	0.0000	20.0000	0
8	1	1	1	-20.0000	0.0000	20.0000	0

EQUATION NUMBERS

NODE NUMBER	DEGREE OF FREEDOM		
N	X	Y	Z
1	1	1	2
2	4	2	3
3	7	5	6
4	2	8	9
5	10	11	12
6	0	0	0
7	0	0	0
8	0	0	0

L O A D   C A S E   D A T A

LOAD CASE NUMBER   = 1
NUMBER OF CONCENTRATED LOADS = 12

NODE NUMBER	DIRECTION	LOAD MAGNITUDE
1	1	-10.0000
1	2	-120.0000
1	3	-105.0000
2	1	-10.0000
2	2	-100.0000
2	3	-105.0000
3	1	10.0000
3	2	-100.0000
3	3	-165.0000
4	1	10.0000
4	2	-120.0000
4	3	-165.0000

E L E M E N T   G R O U P   D A T A

E L E M E N T   D E F I N I T I O N

TRUSS ELEMENT TYPE   NPAR(1)= 1
NUMBER OF ELEMENTS IN THIS ELEMENT GROUP = 12

MATERIAL DEFINITION

NUMBER OF MATERIAL SETS AND CROSS-SEC. AREAS= 1

SET NUMBER	YOUNG'S MODULUS E	CROSS-SEC. AREA A
1	10.00E+06	.28

ELEMENT INFORMATION

ELEMENT NUMBER-N	NODE I 1	NODE J 2	MATERIAL SET NUMBER 1
2	2	3	1
3	3	4	1
4	4	1	1
5	1	8	1
6	2	5	1
7	3	6	1
8	4	7	1
9	5	4	1
10	6	1	1
11	7	2	1
12	2	3	1

MESH PLOT
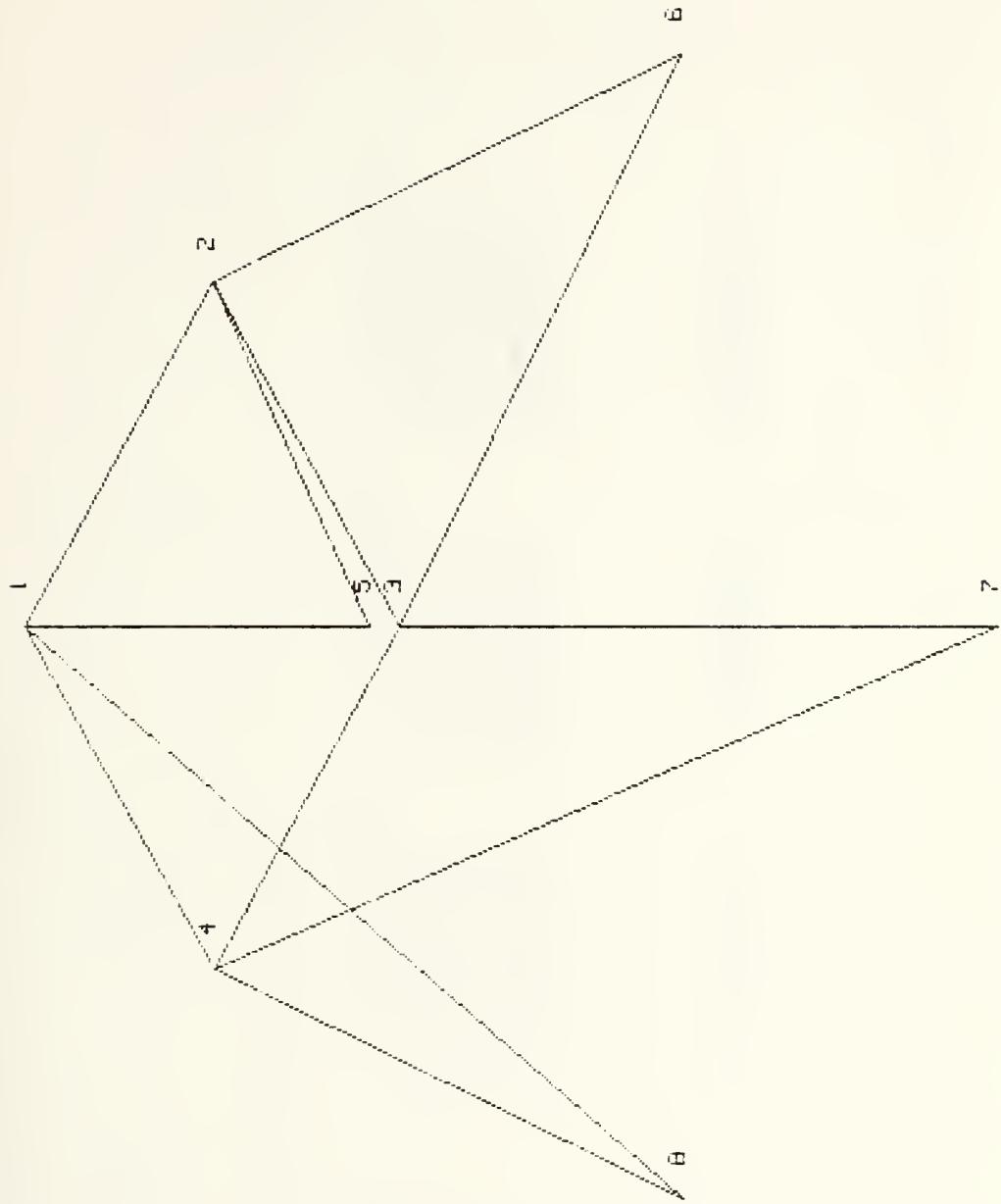
TOTAL SYSTEM DATA

NUMBER OF EQUATIONS   (NEQ) = 12
NUMBER OF MATRIX ELEMENTS (NWK) = 69
MAXIMUM HALF BANDWIDTH   (MK) = 12
HALF BANDWIDTH   (MM) = 5

LOAD CASE 1

D I S P L A C E M E N T S

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	-72.143699E-05	-21.556575E-04	-85.056686E-04
2	-58.286552E-05	86.309128E-05	-70.189897E-04
3	-63.239256E-05	-52.067843E-05	-63.475611E-04
4	-48.953542E-05	-38.089446E-04	-10.194240E-03
5	00.000000E-01	00.000000E-01	00.000000E-01
6	00.000000E-01	00.000000E-01	00.000000E-01
7	00.000000E-01	00.000000E-01	00.000000E-01
8	00.000000E-01	00.000000E-01	00.000000E-01

STRESS CALCULATIONS FOR ELEMENT GROUP 1

| ELEMENT NUMBER	FORCE	STRESS
1	-2.200000E+01	-7.857143E+01
2	7.833333E+01	2.797619E+02
3	-1.666667E+01	-5.952381E+01
4	-1.970000E+02	-7.035714E+02
5	3.009622E+02	1.074865E+03
6	-1.634856E+01	-5.838773E+01
7	-3.009622E+02	-1.074865E+03
8	-2.823843E+01	-1.008515E+02
9	-1.079435E+02	-3.855127E+02
10	-3.446712E+02	-1.230969E+03
11	-9.511857E+01	-3.397092E+02
12	1.095467E+02	3.912381E+02

NORMAL TERMINATION
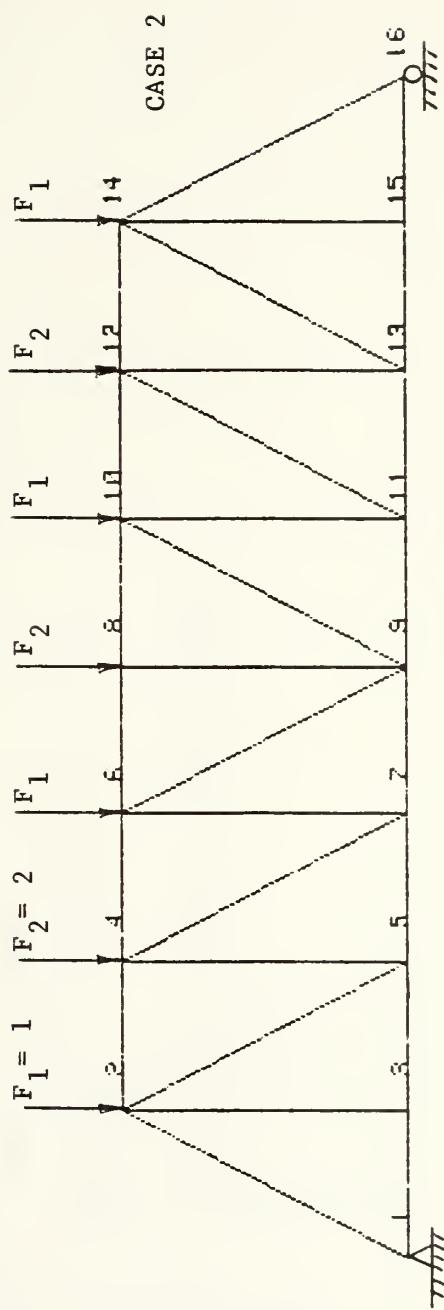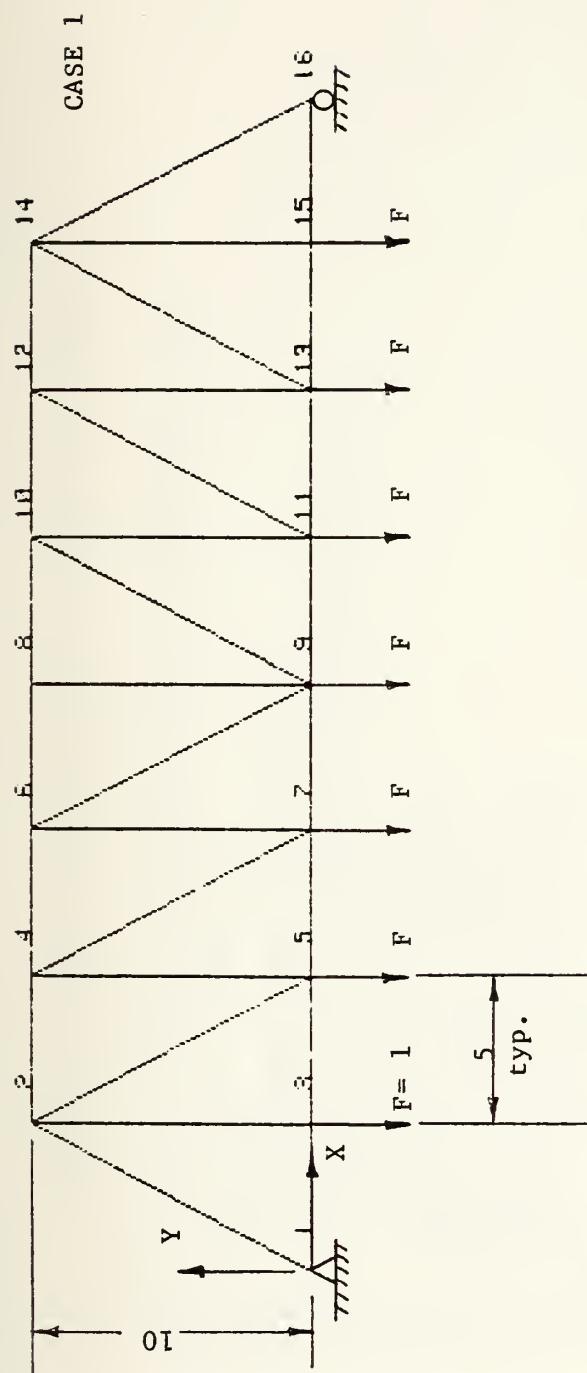
Figure 8. 2-D Truss Example

61

Table VI. Program "INPUT" Listing for 2-D Truss Example

```
10    ! TITLE, CONTROL, BOUNDARY, AND NODE COORDINATES DATA
20    !
30    ASSIGN #1 TO "COOR:T14"
40    READ #1,1
50    ! HED#,NUMNP,NUMEG,NLCASE,MODEX
60    PRINT #1;"ANOTHER EXAMPLE TRUSS PROBLEM",16,1,2,0
70    ! N,ID(1,N),ID(2,N),ID(3,N),X(N),Y(N),Z(N),KN
80    ! FOR N=1,2,14,3,15,16
90    PRINT #1;1,1,1,1,0,0,0,0
100   PRINT #1;2,0,0,1,5,10,0,2
110   PRINT #1;14,0,0,1,35,10,0,0
120   PRINT #1;3,0,0,1,5,0,0,2
130   PRINT #1;15,0,0,1,35,0,0,0
140   PRINT #1;16,0,1,1,40,0,0,0
150   !
160   ! LOADING AT NODES
170   !
180   ASSIGN #1 TO "LOADS:T14"
190   READ #1,1
200   ! LL,NLOAD
210   PRINT #1;1,7
220   ! NOD,IDIRN,FLOAD FOR 7 NODE NUMBERS
230   PRINT #1;3,2,-1
240   PRINT #1;5,2,-1
250   PRINT #1;7,2,-1
260   PRINT #1;9,2,-1
270   PRINT #1;11,2,-1
280   PRINT #1;13,2,-1
290   PRINT #1;15,2,-1
```

```
300 ! LL,NLOAD
310 PRINT #1;2,7
320 ! NOD,IDIRN,FLOAD FOR 7 NODE NUMBERS
330 PRINT #1;2,2,-1
340 PRINT #1;4,2,-2
350 PRINT #1;6,2,-1
360 PRINT #1;8,2,-2
370 PRINT #1;10,2,-1
380 PRINT #1;12,2,-2
390 PRINT #1;14,2,-1
400 !
410 ! MATERIAL, ELEMENT CONNECTIVITY FOR ELEMENT GROUPS
420 !
430 ASSIGN #1 TO "TRUSS:T14"
440 READ #1,1
450 ! NPAR(1),NPAR(2),NPAR(3)
460 PRINT #1;1,29,1
470 ! N,YCH),AREACH)
480 PRINT #1;1,30000,1
490 ! M,II,JJ,MTYPE,KG
500 ! FOR M=1,8,9,15,22,23,26,29
510 PRINT #1;1,1,3,1,2
520 PRINT #1;8,15,16,1,0
530 PRINT #1;9,2,4,1,2
540 PRINT #1;15,2,3,1,2
550 PRINT #1;22,1,2,1,0
560 PRINT #1;23,2,5,1,2
570 PRINT #1;26,9,10,1,2
580 PRINT #1;29,14,16,1,1
590 END
```

Table VII. Data Output for 2-D Truss Example

ANOTHER EXAMPLE TRUSS PROBLEM

C O N T R O L   I N F O R M A T I O N

NUMBER OF NODAL POINTS   (NUMNP) = 16
NUMBER OF ELEMENT GROUPS (NUMEG) = 1
NUMBER OF LOAD CASES   (NLCASE) = 2
SOLUTION MODE   (MODEX) = 0
   EQ. 0, DATA CHECK
   EQ. 1, EXECUTION

N O D A L   P O I N T   D A T A

INPUT NODAL DATA

NODE NUMBER	BOUNDARY CONDITION CODES			NODAL POINT COORDINATES			MESH GENERATING CODE
	X	Y	Z	X	Y	Z	KN
1	1	1	1	0.0000	0.0000	0.0000	0
2	0	0	1	5.0000	10.0000	0.0000	2
14	0	0	1	35.0000	10.0000	0.0000	2
3	0	0	1	5.0000	0.0000	0.0000	2
15	0	1	1	35.0000	0.0000	0.0000	0
16	0	1	1	40.0000	0.0000	0.0000	0

GENERATED NODAL DATA

NODE NUMBER	BOUNDARY CONDITION CODES			NODAL POINT COORDINATES			MESH GENERATING CODE
	X	Y	Z	X	Y	Z	KH
1	1	1	1	0.0000	0.0000	0.0000	0
2	0	0	1	5.0000	10.0000	0.0000	0
3	0	0	1	5.0000	0.0000	0.0000	0
4	0	0	1	10.0000	10.0000	0.0000	0
5	0	0	1	10.0000	0.0000	0.0000	0
6	0	0	1	15.0000	10.0000	0.0000	0
7	0	0	1	15.0000	0.0000	0.0000	0
8	0	0	1	20.0000	10.0000	0.0000	0
9	0	0	1	20.0000	0.0000	0.0000	0
10	0	0	1	25.0000	10.0000	0.0000	0
11	0	0	1	25.0000	0.0000	0.0000	0
12	0	0	1	30.0000	10.0000	0.0000	0
13	0	0	1	30.0000	0.0000	0.0000	0
14	0	0	1	35.0000	10.0000	0.0000	0
15	0	0	1	35.0000	0.0000	0.0000	0
16	0	1	1	40.0000	0.0000	0.0000	0

EQUATION NUMBERS

NODE NUMBER N	DEGREE OF FREEDOM		
	X	Y	Z
1	0	0	0
2	1	2	0
3	3	4	0
4	5	6	0
5	7	8	0
6	9	10	0
7	11	12	0
8	13	14	0
9	15	16	0
10	17	18	0
11	19	20	0
12	21	22	0
13	23	24	0
14	25	26	0
15	27	28	0
16	29	0	0

L O A D   C A S E   D A T A

LOAD CASE NUMBER   = 1
NUMBER OF CONCENTRATED LOADS = 7

NODE NUMBER	DIRECTION	LOAD MAGNITUDE
3	2	-1.0000
5	2	-1.0000
7	2	-1.0000
9	2	-1.0000
11	2	-1.0000
13	2	-1.0000
15	2	-1.0000

LOAD CASE NUMBER   = 2
NUMBER OF CONCENTRATED LOADS = 7

NODE NUMBER	DIRECTION	LOAD MAGNITUDE
2	2	-1.0000
4	2	-2.0000
6	2	-1.0000
8	2	-2.0000
10	2	-1.0000
12	2	-2.0000
14	2	-1.0000

ELEMENT GROUP DATA

ELEMENT DEFINITION

TRUSS ELEMENT TYPE   NPAR(1)= 1
NUMBER OF ELEMENTS IN THIS ELEMENT GROUP = 29

MATERIAL DEFINITION

NUMBER OF MATERIAL SETS AND CROSS-SEC. AREAS= 1

SET   YOUNG'S   CROSS-SEC.
NUMBER   MODULUS   AREA
         E   A

  1   30.00E+03   1.00

ELEMENT INFORMATION

ELEMENT NUMBER-N	NODE	NODE	MATERIAL SET NUMBER
1	1	3	1
2	3	5	1
3	5	7	1
4	7	9	1
5	9	11	1
6	11	13	1
7	13	15	1
8	15	16	1
9	2	4	1
10	4	6	1
11	6	8	1
12	8	10	1
13	10	12	1
14	12	14	1
15	14	16	1
16	1	4	1
17	2	6	1
18	4	8	1
19	6	10	1
20	8	12	1
21	10	14	1
22	12	16	1
23	1	10	1
24	2	12	1
25	4	14	1
26	6	16	1
27	9		1
28	11		1
29	13		1

69

MESH PLOT

TOTAL SYSTEM DATA

NUMBER OF EQUATIONS   (NEQ) = 29
NUMBER OF MATRIX ELEMENTS   (NMK) = 159
MAXIMUM HALF BANDWIDTH   (MK) = 8
HALF BANDWIDTH   (MM) = 5

MODEX=0 OVERRIDDEN. PROGRAM CONTINUES...

LOAD CASE 1

D I S P L A C E M E N T S

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	00.0000000E-01	00.0000000E-01	00.0000000E-01
2	35.0000000E-04	-33.8046662E-04	00.0000000E-01
3	29.1666667E-05	-37.1379960E-04	00.0000000E-01
4	30.0000000E-04	-65.0341830E-04	00.0000000E-01
5	58.3333330E-05	-60.0341830E-04	00.0000000E-01
6	23.7500000E-04	-83.2718950E-04	00.0000000E-01
7	10.8333330E-04	-81.6052290E-04	00.0000000E-01
8	17.0833330E-04	-88.9344660E-04	00.0000000E-01
9	17.0833330E-04	-88.9344660E-04	00.0000000E-01
10	10.4166667E-04	-83.2718950E-04	00.0000000E-01
11	23.3333330E-04	-81.6052290E-04	00.0000000E-01
12	41.6666667E-05	-65.0341830E-04	00.0000000E-01
13	28.3333330E-04	-60.0341830E-04	00.0000000E-01
14	-83.3333330E-06	-33.8046662E-04	00.0000000E-01
15	31.2500000E-04	-37.1379960E-04	00.0000000E-01
16	34.1666667E-04	00.0000000E-01	00.0000000E-01

ELEMENT NUMBER	FORCE	STRESS
1	1.750000E+00	1.750000E+00
2	1.750000E+00	1.750000E+00
3	3.000000E+00	3.000000E+00
4	3.750000E+00	3.750000E+00
5	3.750000E+00	3.750000E+00
6	3.000000E+00	3.000000E+00
7	1.750000E+00	1.750000E+00
8	-1.750000E+00	-1.750000E+00
9	-3.000000E+00	-3.000000E+00
10	-3.750000E+00	-3.750000E+00
11	-4.000000E+00	-4.000000E+00
12	-4.000000E+00	-4.000000E+00
13	-3.750000E+00	-3.750000E+00
14	-3.000000E+00	-3.000000E+00
15	-1.000000E+00	-1.000000E+00
16	-1.500000E+00	-1.500000E+00
17	-5.000000E-01	-5.000000E-01
18	0.000000E+00	0.000000E+00
19	-5.000000E-01	-5.000000E-01
20	-1.500000E+00	-1.500000E+00
21	-1.000000E+00	-1.000000E+00
22	-3.913119E+00	-3.913119E+00
23	2.795085E+00	2.795085E+00
24	1.677051E+00	1.677051E+00
25	5.590170E-01	5.590170E-01
26	5.590170E-01	5.590170E-01
27	-1.677051E+00	-1.677051E+00
28	2.795085E+00	2.795085E+00
29	-3.913119E+00	-3.913119E+00

DISPLACEMENTS

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	00.0000000E-01	00.0000000E-01	00.0000000E-01
2	51.666662E-04	-49.125708E-04	00.0000000E-01
3	41.666662E-05	-49.125708E-04	00.0000000E-01
4	44.166662E-04	-10.275961E-03	00.0000000E-01
5	33.333333E-05	-09.426275E-04	00.0000000E-01
6	35.000000E-04	-19.230089E-03	00.0000000E-01
7	15.833333E-04	-12.624322E-03	00.0000000E-01
8	25.000000E-04	-14.923503E-03	00.0000000E-01
9	25.000000E-04	-14.256837E-03	00.0000000E-01
10	15.000000E-04	-13.290989E-03	00.0000000E-01
11	34.166662E-04	-12.624322E-03	00.0000000E-01
12	50.333333E-05	-10.275761E-03	00.0000000E-01
13	41.666662E-04	-09.426275E-04	00.0000000E-01
14	-16.666662E-05	-49.125708E-04	00.0000000E-01
15	45.833333E-04	-49.125708E-04	00.0000000E-01
16	50.000000E-04	00.0000000E-01	00.0000000E-01

ELEMENT NUMBER	FORCE	STRESS
1	2.5000000E+00	2.5000000E+00
2	2.5000000E+00	2.5000000E+00
3	4.5000000E+00	4.5000000E+00
4	5.5000000E+00	5.5000000E+00
5	5.5000000E+00	5.5000000E+00
6	-4.5000000E+00	-4.5000000E+00
7	2.5000000E+00	2.5000000E+00
8	2.5000000E+00	2.5000000E+00
9	-4.5000000E+00	-4.5000000E+00
10	-5.5000000E+00	-5.5000000E+00
11	-6.0000000E+00	-6.0000000E+00
12	-6.0000000E+00	-6.0000000E+00
13	-5.5000000E+00	-5.5000000E+00
14	-4.5000000E+00	-4.5000000E+00
15	0.0000000E+00	0.0000000E+00
16	-4.0000000E+00	-4.0000000E+00
17	-2.0000000E+00	-2.0000000E+00
18	-2.0000000E+00	-2.0000000E+00
19	-2.0000000E+00	-2.0000000E+00
20	-4.0000000E+00	-4.0000000E+00
21	0.0000000E+00	0.0000000E+00
22	5.5901700E+00	5.5901700E+00
23	4.4721360E+00	4.4721360E+00
24	2.2360680E+00	2.2360680E+00
25	1.1180340E+00	1.1180340E+00
26	1.1180340E+00	1.1180340E+00
27	2.2360680E+00	2.2360680E+00
28	4.4721360E+00	4.4721360E+00
29	-5.5901700E+00	-5.5901700E+00

NORMAL TERMINATION

COMPUTER PROGRAM LISTING

```
10    ! M A I N   P R O G R A M        REVISED:
20    !
30    !
40    OPTION BASE 1
50    COM Ng,Modex,Npar(3),Numop,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Nk,Nlcase
60    Numest=Maxest=0
70    DIM Hed$[50]
80    !
90    ! ******** I N P U T   P H A S E  ********
100   !
110   ASSIGN #1 TO "A1:T14"
120   ASSIGN #2 TO "A2:T14"
130   ASSIGN #3 TO "A3:T14"
140   ASSIGN #4 TO "A4:T14"
150   ASSIGN #5 TO "ILOAD:T14"
160   !
170   ! READ CONTROL INFO
180   !
190   ASSIGN #10 TO "COOR:T14"
200   READ #10,1
210   READ #10;Hed$,Numnp,Numeg,Nlcase,Modex
220   IF Numnp=0 THEN Cont1
230   GOTO Cont2
240 Cont1: PRINT LIN(2),"NUMNP = 0     PROGRAM STOPS..."
250   GOTO Finish
260 Cont2: PRINT TAB(10),Hed$,LIN(2)
270   PRINT LIN(2),"C O N T R O L   I N F O R M A T I O N",LIN(2)
280   PRINT TAB(6),"NUMBER OF NODAL POINTS   (NUMNP)   =";Numnp
290   PRINT TAB(6),"NUMBER OF ELEMENT GROUPS (NUMEG)   =";Numeg
300   PRINT TAB(6),"NUMBER OF LOAD CASES     (NLCASE)  =";Nlcase
```

```
310 PRINT TAB(6),"SOLUTION MODE          (MODEX)     =";Modex
320 PRINT TAB(11),"EQ. 0, DATA CHECK"
330 PRINT TAB(11),"EQ. 1, EXECUTION"
340 N2=N3=N4=Numnp
350 Inputt: CALL Input(N2,N3,N4,#10,#1,#2,#3,#4)
360 Neq1=Neq+1
370 !
380 ! CALCULATE AND STORE LOAD VECTORS
390 !
400 N5=Neq
410 PRINT LIN(2),"L O A D   C A S E     D A T A",LIN(2)
420 READ #5,1
430 ASSIGN #9 TO "LOADS:T14"
440 READ #9,1
450 FOR L=1 TO Nlcase
460 READ #9;L1,Nload
470 PRINT LIN(2),"LOAD CASE NUMBER";SPA(13);"=";L
480 PRINT "NUMBER OF CONCENTRATED LOADS =";Nload
490 IF L1=L THEN L310
500 PRINT LIN(2),"***ERROR...LOAD CASES ARE NOT IN ORDER. PROGRAM STOPS.."
510 GOTO Finish
520 L310: ! CONTINUE
530 N6=N7=N8=Nload
540 CALL Loads(N5,N6,N7,N8,Nload,#5,#9,#1)
550 NEXT L
560 !
570 ! READ, GENERATE, AND STORE ELEMENT DATA
580 !
590 Ind=1
600 CALL Elcal
610 !
620 !******** S O L U T I O N   P H A S E ********
630 !
640 ! ASSEMBLE STIFFNESS MATRIX
650 !
660 GET "SSAPO",3250,Ssapogo
670 Ssapogo: N2=Neq+1
680 N5=Neq
```

```
690   CALL Address(N2,N5)
700   Mm=Nwk/Neq
710   Mm=INT(Mm)
720   N2=Neq1=Neq+1
730   N3=Nwk
740   N4=Neq
750   N5=Maxest
760   !
770   ! WRITE TOTAL SYSTEM DATA
780   !
790   PRINT LIN(2),"TOTAL SYSTEM DATA",LIN(2)
800   PRINT "NUMBER OF EQUATIONS       (NEQ) =";Neq
810   PRINT "NUMBER OF MATRIX ELEMENTS (NWK) =";Nwk
820   PRINT "MAXIMUM HALF BANDWIDTH     (MK) =";Mk
830   PRINT "HALF BANDWIDTH             (MM) =";Mm
840   !
850   ! FOR MODEX=0 (DATA CHECK ONLY), PROGRAM PAUSES HERE.
860   !
870   IF Modex=0 THEN Test
880   GOTO L100
890   Test: DISP "MODEX=0, DATA CHECK. TO OVERRIDE, PRESS CONT KEY, ELSE PRESS S10
P KEY."
900   PAUSE
910   DISP "MODEX RESET TO 1 IN PROGRAM."
920   PRINT LIN(2),"MODEX=0 OVERRIDDEN. PROGRAM CONTINUES...",LIN(2)
930   Modex=1
940   L100: Nnl=Nwk+Neq
950   Ind=2
960   CALL Assem
970   !
980   !   TRIANGULARIZE STIFFNESS MATRIX
990   !
1000  Ktr=1
1010  ASSIGN #8 TO "ILOAD:T14"
1020  READ #8,1
1030  CALL Colsol(N3,N4,N2,Neq1,Ktr,#8)
1040  Ktr=2
1050  Ind=3
1060  FOR L=1 TO Nlcase
```

77

```
1070 !     CALCULATION OF DISPLACEMENTS
1080 !
1090 !
1100 CALL Colsol(H3,H4,N2,Neq1,Ktr,#8)
1110 PRINT LIN(2),"LOAD CASE";L,LIN(2)
1120 CALL Write
1130 !
1140 !     CALCULATION OF STRESSES
1150 !
1160 CALL Stress
1170 NEXT L
1180 Finish:PRINT LIN(2),"NORMAL TERMINATION"
1190 END
1200 SUB Element(#10,#9,#8)
1210 OPTION BASE 1
1220 !
1230 ! ** PROGRAM TO CALL THE APPROPRIATE ELEMENT SUBROUTINE..
1240 !
1250 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
1260 Npar1=Npar(1)
1270 ON Npar1 GOTO L1,L2,L3
1280 L1:CALL Truss(#10,#9,#8)
1290 SUBEXIT
1300 !
1310 ! OTHER ELEMENT TYPES WOULD BE CALLED HERE,  IDENTIFYING EACH
1320 ! ELEMENT TYPE BY A DIFFERENT NPAR(1) VALUE...
1330 !
1340 L2:SUBEXIT
1350 L3:SUBEXIT
1360 SUBEND
1370 SUB Truss(#10,#9,#8)
1380 OPTION BASE 1
1390 !
1400 ! ** PROGRAM TO SET UP STORAGE AND CALL THE TRUSS ELEMENT SUBROUTINE...
1410 !
1420 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
1430 Nume=Npar(2)
1440 Nummat=Npar(3)
```

78

```
1450 N2=N3=N4=Numrp
1460 N5=Neq
1470 IF Ind=1 THEN L100
1480 N2=Neq+1
1490 N3=Nwk
1500 N4=Neq
1510 L100:N101=N102=Nummat
1520 N103=N104=6*Nume
1530 N105=Nume
1540 Midest=N101+N102+N103+N104+N105
1550 CALL Russ(N2,N3,N4,N5,N101,N102,Nume,N105,#10,#9,#8)
1560 SUBEND
1570 SUB Russ(N2,N3,N4,N5,N101,N102,Nume,N105,#10,#9,#8)
1580 OPTION BASE 1
1590 !
1600 ! ** TRUSS ELEMENT SUBROUTINE...
1610 !
1620 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
1630 DIM X(N2),Y(N3),Z(N3),Id(3,Numnp),E(N101),Area(N102),Lm(6,Nume)
1640 DIM Xyz(6,Nume),Matp(N105),U(N4),Mht(N5)
1650 DIM S(21),St(6),D(3),Lim(10),A(N3),Maxa(N2)
1660 Npar1=Npar(1)
1670 Nume=Npar(2)
1680 Nummat=Npar(3)
1690 Nd=6
1700 ON Ind GOTO L300,L610,L900
1710 !
1720 ! R E A D   A N D   G E N E R A T E   E L E M E N T   I N F O
1730 !
1740 ! READ MATERIAL INFO
1750 !
1760 L300:ASSIGN #1 TO "A1:T14"
1770 ASSIGN #2 TO "A2:T14"
1780 ASSIGN #3 TO "A3:T14"
1790 ASSIGN #4 TO "A4:T14"
1800 READ #1,1
1810 READ #2,1
1820 READ #3,1
```

```
1830 READ #4,1
1840 MAT READ #1;Id
1850 MAT READ #2;X
1860 MAT READ #3;Y
1870 MAT READ #4;Z
1880 PRINT LIN(2),"E L E M E N T   D E F I N I T I O N",LIN(2)
1890 PRINT TAB(6),"TRUSS ELEMENT TYPE";SPA(16);"NPAR(1)= 1"
1900 PRINT TAB(6),"NUMBER OF ELEMENTS IN THIS ELEMENT GROUP =";Nume
1910 IF Nummat=0 THEN Nummat=1
1920 PRINT LIN(2),"M A T E R I A L   D E F I N I T I O N",LIN(2)
1930 PRINT TAB(6),"NUMBER OF MATERIAL SETS AND CROSS-SEC. AREAS=";Nummat
1940 PRINT LIN(2)," SET";SPA(6);"YOUNG'S";SPA(5);"CROSS-SEC."
1950 PRINT " NUMBER";SPA(5);"MODULUS";SPA(10);"AREA"
1960 PRINT SPA(15);"E";SPA(14);"A"
1970 FOR I=1 TO Nummat
1980 READ #10;N,E(N),Area(N)
1990 PRINT USING Pr1;N,E(N),Area(N)
2000 NEXT I
2010 !
2020 ! READ ELEMENT INFORMATION
2030 !
2040 PRINT LIN(2),"E L E M E N T   I N F O R M A T I O N",LIN(2)
2050 PRINT " ELEMENT";SPA(5);"NODE";SPA(5);"NODE";SPA(7);"MATERIAL"
2060 PRINT " NUMBER-N";SPA(6);"I";SPA(8);"J";SPA(7);"SET NUMBER"
2070 N=1
2080 L100:READ #10;M,Ii,Jj,Mtyp,Kg
2090 IF Kg=0 THEN Kg=1
2100 L120:IF M<>N THEN L200
2110 I=Ii
2120 J=Jj
2130 Mtype=Mtyp
2140 Kkk=Kg
2150 !
2160 ! SAVE ELEMENT INFORMATION
2170 !
2180 L200:Xyz(1,N)=X(I)
2190 Xyz(2,N)=Y(I)
2200 Xyz(3,N)=Z(I)
```

```
2210 Xyz(4,N)=X(J)
2220 Xyz(5,N)=Y(J)
2230 Xyz(6,N)=Z(J)
2240 Matp(N)=Mtype
2250 FOR L=1 TO 6
2260 Lm(L,N)=0
2270 NEXT L
2280 FOR L=1 TO 3
2290 Lm(L)=Lm(L,N)=Id(L,I)
2300 Lm(L+3)=Lm(L+3,N)=Id(L,J)
2310 NEXT L
2320 !
2330 ! UPDATE COLUMN HEIGHTS AND BANDWIDTH
2340 !
2350 CALL Colht(Mht(*),Nd,Llm(*))
2360 PRINT USING Pr2;N,I,J,Mtype
2370 IF N=Nume THEN Lexit
2380 N=N+1
2390 I=I+Kkk
2400 J=J+Kkk
2410 IF N>M THEN L100
2420 GOTO L120
2430 Lexit:ASSIGN #5 TO "A5:T14"
2440 READ #5,1
2450 MAT PRINT #5;Mht
2460 PRINT #9;Midest,Npar(1),Npar(2),Npar(3)
2470 MAT PRINT #9;E,Area,Lm,Xyz,Matp
2480 PRINT #8;Nume
2490 MAT PRINT #8;Xyz
2500 SUBEXIT
2510 !
2520 !A S S E M B L E   S T R U C T U R E   S T I F F N E S S   M A T R I X
2530 !
2540 L610:MAT READ #10;E,Area,Lm,Xyz,Matp
2550 ASSIGN #2 TO "A2:T14"
2560 READ #2,1
2570 MAT READ #2;Maxa
2580 FOR N=1 TO Nume
```

```
2590 Mtype=Matp(N)
2600 X12=0
2610 FOR L=1 TO 3
2620 D(L)=Xyz(L,N)-Xyz(L+3,N)
2630 X12=X12+D(L)*D(L)
2640 NEXT L
2650 X1=SQR(X12)
2660 Xx=E(Mtype)*Area(Mtype)*X1
2670 FOR L=1 TO 3
2680 St(L)=D(L)/X12
2690 St(L+3)=-St(L)
2700 NEXT L
2710 K1=0
2720 FOR L=1 TO 6
2730 Yy=St(L)*Xx
2740 FOR K=L TO 6
2750 K1=K1+1
2760 S(K1)=St(K)*Yy
2770 NEXT K
2780 NEXT L
2790 FOR I=1 TO Nd
2800 Lim(I)=Lm(I,N)
2810 NEXT I
2820 CALL Addban(A(*),Maxa(*),S(*),Lim(*),Nd)
2830 NEXT N
2840 ASSIGN #3 TO "A3:T14"
2850 READ #3,1
2860 MAT PRINT #3;A
2870 SUBEXIT
2880 !
2890 !  S T R E S S   C A L C U L A T I O N S
2900 !
2910 L900:ASSIGN #4 TO "A4:T14"
2920 READ #4,1
2930 MAT READ #4;U
2940 MAT READ #10;E,Area,Lm,Xyz,Matp
2950 PRINT LIN(2),"S T R E S S   C A L C U L A T I O N S   F O R   E L E M E
N T   G R O U P";Ng,LIN(2)
2960 PRINT "  ELEMENT";SPA(12);"FORCE";SPA(12);"STRESS"
```

```
2970 PRINT "    NUMBER"
2980 FOR N=1 TO Nume
2990 Mtype=Matp(N)
3000 X12=0
3010 FOR L=1 TO 3
3020 D(L)=Xyz(L,N)-Xyz(L+3,N)
3030 X12=X12+D(L)*D(L)
3040 NEXT L
3050 FOR L=1 TO 3
3060 St(L)=D(L)/X12*E(Mtype)
3070 St(L+3)=-St(L)
3080 NEXT L
3090 Str=0
3100 FOR L=1 TO 3
3110 I=Lm(L,N)
3120 IF I<=0 THEN L807
3130 Str=Str+St(L)*U(I)
3140 L807:J=Lm(L+3,N)
3150 IF J<=0 THEN L806
3160 Str=Str+St(L+3)*U(J)
3170 L806:NEXT L
3180 P=Str*Area(Mtype)
3190 PRINT USING Pr3;N,P,Str
3200 NEXT N
3210 Pr1:IMAGE 3X,DD,6X,DD.DDE,2X,9D.2D
3220 Pr2:IMAGE 5D,6X,5D,4X,5D,7X,5D
3230 Pr3:IMAGE 6D,11X,MD.6DE,4X,MD.6DE
3240 SUBEND
3250 ! SSAPO OVERLAYED HERE.
3260 SUB Elcal
3270 OPTION BASE 1
3280 !
3290 ! ** PROGRAM TO LOOP OVER ALL ELEMENT GROUPS FOR READING,
3300 !    GENERATING AND STORING THE ELEMENT DATA...
3310 !
3320 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,N1case
3330 PRINT LIN(2),"E L E M E N T   G R O U P   D A T A"
3340 !
```

```
3350 ! LOOP OVER ALL ELEMENT GROUPS
3360 !
3370 ASSIGN #8 TO "MESH:T14"
3380 ASSIGN #9 TO "IELMNT:T14"
3390 ASSIGN #10 TO "TRUSS:T14"
3400 READ #8,1
3410 READ #9,1
3420 READ #10,1
3430 FOR N=1 TO Numeg
3440 READ #10;Npar(1),Npar(2),Npar(3)
3450 CALL Elemnt(#10,#9,#8)
3460 IF Midest>Maxest THEN Maxest=Midest
3470 NEXT N
3480 !
3490 ! PLOT MESH
3500 !
3510 READ #8,1
3520 Xfile$="A2:T14"
3530 Yfile$="A3:T14"
3540 Zfile$="A4:T14"
3550 FOR N=1 TO Numeg
3560 READ #8;Nume
3570 CALL Pmesh(N,Nume,#8,Xfile$,Yfile$,Zfile$,Flag)
3580 NEXT N
3590 SUBEND
3600 SUB Colht(Mht(*),Nd,Lm(*))
3610 OPTION BASE 1
3620 !
3630 ! ** PROGRAM TO CALCULATE COLUMN HEIGHTS
3640 !
3650 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Nk,Nlcase
3660 REDIM Lm(Nd)
3670 Ls=100000
3680 FOR I=1 TO Nd
3690 IF Lm(I)<>0 THEN L110
3700 IF Lm(I)=0 THEN L100
3710 L110:IF Lm(I)-Ls>=0 THEN L100
3720 IF Lm(I)-Ls<0 THEN L120
```

```
3730 L120:Ls=Lm(I)
3740 L100:NEXT I
3750 FOR I=1 TO Nd
3760 Ii=Lm(I)
3770 IF Ii=0 THEN L200
3780 Me=Ii-Ls
3790 IF Me>Mht(Ii) THEN Mht(Ii)=Me
3800 L200:NEXT I
3810 SUBEND
3820 SUB Input(N2,N3,N4,#10,#1,#2,#3,#4)
3830 !
3840 ! **PROGRAM TO READ,GENERATE AND PRINT NODAL POINT INPUT DATA
3850 ! TO CALCULATE EQUATION NUMBERS AND STORE THEM IN ID ARRAY ***
3860 !
3870 OPTION BASE 1
3880 COM Ng,Modex,Npar(3),Numnp,Neq,Hwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
3890 DIM X(N2),Y(N3),Z(N4),Id(3,Numnp)
3900 !
3910 ! READ AND GENERATE NODAL POINT DATA
3920 !
3930 PRINT LIN(2),"N O D A L    P O I N T      D A T A",LIN(2)
3940 PRINT "INPUT NODAL DATA"
3950 PRINT LIN(1)
3960 PRINT USING Pr1
3970 PRINT USING Pr2
3980 PRINT USING Pr3
3990 PRINT USING Pr4
4000 PRINT LIN(1)
4010 Knold=Nold=0
4020 L10:READ #10;N,Id(1,N),Id(2,N),Id(3,N),X(N),Y(N),Z(N),Kn
4030 PRINT USING Prx;N,Id(1,N),Id(2,N),Id(3,N),X(N),Y(N),Z(N),Kn
4040 IF Knold=0 THEN L50
4050 Numm=(N-Nold)/Knold
4060 Numm=Numm-1
4070 IF Numm<1 THEN L50
4080 Xnum=Numm
4090 Dx=(X(N)-X(Nold))/Xnum
4100 Dy=(Y(N)-Y(Nold))/Xnum
```

```
4110 Dz=(Z(N)-Z(Nold))/Xnum
4120 K=Nold
4130 FOR J=1 TO Numn
4140 Kk=K
4150 K=K+Knold
4160 X(K)=X(Kk)+Dx
4170 Y(K)=Y(Kk)+Dy
4180 Z(K)=Z(Kk)+Dz
4190 FOR I=1 TO 3
4200 Id(I,K)=Id(I,Kk)
4210 NEXT I
4220 NEXT J
4230 L50:Nold=N
4240 Knold=Kn
4250 IF N<>Numnp THEN L10
4260 !
4270 ! WRITE COMPLETE NODAL DATA
4280 !
4290 PRINT LIN(2),"GENERATED NODAL DATA",LIN(1)
4300 PRINT USING Pr1
4310 PRINT USING Pr2
4320 PRINT USING Pr3
4330 PRINT USING Pr4
4340 PRINT LIN(1)
4350 FOR N=1 TO Numnp
4360 PRINT USING Prx;N,Id(1,N),Id(2,N),Id(3,N),X(N),Y(N),Z(N),Kn
4370 NEXT N
4380 READ #2,1
4390 MAT PRINT #2;X
4400 READ #3,1
4410 MAT PRINT #3;Y
4420 READ #4,1
4430 MAT PRINT #4;Z
4440 !
4450 ! NUMBER THE UNKNOWNS
4460 !
4470 Neq=0
4480 FOR N=1 TO Numnp
```

86

```
4490 FOR I=1 TO 3
4500 IF Id(I,N)<>0 THEN L110
4510 IF Id(I,N)=0 THEN L120
4520 L120:Neq=Neq+1
4530 Id(I,N)=Neq
4540 GOTO L100
4550 L110:Id(I,N)=0
4560 L100:NEXT I
4570 NEXT N
4580 !
4590 ! WRITE EQUATION NUMBERS
4600 !
4610 PRINT LIN(2),"EQUATION NUMBERS"
4620 PRINT LIN(1)
4630 PRINT USING Pr5
4640 PRINT "   NUMBER"
4650 PRINT USING Pr6
4660 FOR N=1 TO Numnp
4670 PRINT USING Pr7;N,Id(1,N),Id(2,N),Id(3,N)
4680 NEXT N
4690 READ #1,1
4700 MAT PRINT #1;Id
4710 Pr1:IMAGE 2X,"NODE",8X,"BOUNDARY",21X,"NODAL POINT",20X,"MESH"
4720 Pr2:IMAGE 1X,"NUMBER",3X,"CONDITION   CODES",17X,"COORDINATES",14X,"GENERATI
NG"
4730 Pr3:IMAGE 24X,"CODE"
4740 Pr4:IMAGE 13X,"X",4X,"Y",4X,"Z",11X,"X",13X,"Y",13X,"Z",11X,"KN"
4750 Pr5:IMAGE 4X,"NODE",8X,"DEGREE OF FREEDOM"
4760 Pr6:IMAGE 5X,"N",13X,"X",4X,"Y",4X,"Z"
4770 Pr7:IMAGE 3X,3D,11X,2(3D,2X),3D
4780 Prx:IMAGE 3X,3D,7X,2(D,4X),D,7X,2(5D.4D,4X),5D.4D,6X,DD
4790 SUBEND
4800 SUB Loads(N5,N6,N7,N8,Nload,#5,#9,#1)
4810 !
4820 !**PROGRAM TO READ NODAL LOAD DATA AND TO CALCULATE THE LOAD VECTOR R
4830 !   FOR EACH LOAD CASE...
4840 !
4850 OPTION BASE 1
4860 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
```

```
4870  DIM R(Neq),Nod(N6),Idirn(N7),Fload(N8),Id(3,Numnp)
4880  PRINT LIN(2);SPA(4);"NODE";SPA(7);"DIRECTION";SPA(7);"LOAD"
4890  PRINT "     NUMBER";SPA(7);"MAGNITUDE"
4900  READ #1,1
4910  MAT READ #1;Id
4920  FOR I=1 TO Nload
4930  READ #9;Nod(I),Idirn(I),Fload(I)
4940  PRINT USING Pri;Nod(I),Idirn(I),Fload(I)
4950  NEXT I
4960  MAT R=ZER
4970  FOR L=1 TO Nload
4980  Ln=Nod(L)
4990  Li=Idirn(L)
5000  Ii=Id(Li,Ln)
5010  IF Ii<=0 THEN L220
5020  IF Ii>0 THEN L240
5030  L240:R(Ii)=R(Ii)+Fload(L)
5040  L220:NEXT L
5050  MAT PRINT #5;R
5060  Pri:IMAGE 3X,3D,11X,3D,6X,8D.4D
5070  SUBEND
5080  SUB Pmesh(N,Nume,#8,Xfile$,Yfile$,Zfile$,Flag)
5090  OPTION BASE 1
5100  !
5110  !   ** PROGRAM TO PLOT MESH DATA
5120  !
5130  COM Ng,Modex,Npar(3),Numnp,Neq,Nut,Ind,Numeg,Numest,Maxest,Midest,Mk
5140  DIM Xyz(6,Nume),X(Numnp),Y(Numnp),Z(Numnp)
5150  PLOTTER IS 13,"GRAPHICS"
5160  GRAPHICS
5170  LOCATE 10,110,0,95
5180   CSIZE 2.5
5190  IF N>1 THEN Lconn
5200  ASSIGN #2 TO Xfile$
5210  ASSIGN #3 TO Yfile$
5220   ASSIGN #4 TO Zfile$
5230  READ #2,1
5240  MAT READ #2;X
```

```
5250 READ #3,1
5260 MAT READ #3;Y
5270    READ #4,1
5280    MAT READ #4;Z
5290 FOR K=1 TO Numnp
5300 IF Z(K)><0 THEN Lgo
5310 NEXT K
5320 ! CONTINUE FOR 2-D PLOT
5330 Xmin=X(1)
5340 Ymin=Y(1)
5350 Xmax=X(1)
5360 Ymax=Y(1)
5370 FOR I=2 TO Numnp
5380 IF X(I)>Xmax THEN Xmax=X(I)
5390 IF Y(I)>Ymax THEN Ymax=Y(I)
5400 IF X(I)<Xmin THEN Xmin=X(I)
5410 IF Y(I)<Ymin THEN Ymin=Y(I)
5420 NEXT I
5430 IF Xmax-Xmin>=Ymax-Ymin THEN L1
5440 SCALE Xmin,Xmin+Ymax-Ymin,Ymin,Ymax
5450 GOTO L2
5460 L1:SCALE Xmin,Xmax,Ymin,Ymin+Xmax-Xmin
5470 L2:FOR K=1 TO Numnp
5480 PLOT X(K),Y(K),-2
5490 IPLOT 0,0,-2
5500 LABEL USING "3D";K
5510 NEXT K
5520 GOTO Lconn
5530 Lgo: ! CONTINUE FOR 3-D PLOT
5540 Flag=3
5550 Xmax=Xmin=.866*(X(1)-Z(1))
5560 Ymax=Ymin=Y(1)-.5*(X(1)+Z(1))
5570 FOR I=2 TO Numnp
5580 IF .866*(X(I)-Z(I))>Xmax THEN Xmax=.866*(X(I)-Z(I))
5590 IF Y(I)-.5*(X(I)+Z(I))>Ymax THEN Ymax=Y(I)-.5*(X(I)+Z(I))
5600 IF .866*(X(I)-Z(I))<Xmin THEN Xmin=.866*(X(I)-Z(I))
5610 IF Y(I)-.5*(X(I)+Z(I))<Ymin THEN Ymin=Y(I)-.5*(X(I)+Z(I))
5620 NEXT I
```

```
5630    IF Xmax-Xmin>=Ymax-Ymin THEN L1a
5640    SCALE Xmin,Xmin+Ymax-Ymin,Ymin,Ymax
5650    GOTO L2a
5660 L1a:    SCALE Xmin,Xmax,Ymin,Ymin+Xmax-Xmin
5670 L2a:    FOR K=1 TO Numnp
5680    Xk=.866*(X(K)-Z(K))
5690    Yk=Y(K)-.5*(X(K)+Z(K))
5700    PLOT Xk,Yk,-2
5710    IPLOT 0,0,-2
5720    LABEL USING "3D";K
5730    NEXT K
5740 Lconn:MAT READ #8;Xyz
5750    IF Flag=3 THEN L20
5760    FOR K=1 TO Nume
5770    PLOT Xyz(1,K),Xyz(2,K),-2
5780    PLOT Xyz(4,K),Xyz(5,K),-1
5790    NEXT K
5800    GOTO L21
5810 L20:    FOR K=1 TO Nume
5820    Xk1=.866*(Xyz(1,K)-Xyz(3,K))
5830    Yk1=Xyz(2,K)-.5*(Xyz(1,K)+Xyz(3,K))
5840    Xk2=.866*(Xyz(4,K)-Xyz(6,K))
5850    Yk2=Xyz(5,K)-.5*(Xyz(4,K)+Xyz(6,K))
5860    PLOT Xk1,Yk1,-2
5870    PLOT Xk2,Yk2,-1
5880    NEXT K
5890 L21:IF N=Numeg THEN L10
5900    GOTO L25
5910 L10:PRINT LIN(2),PAGE
5920    PRINT "MESH PLOT"
5930    DUMP GRAPHICS
5940    GCLEAR
5950    EXIT GRAPHICS
5960    GOTO L30
5970 L25:EXIT GRAPHICS
5980 L30:!
5990    SUBEND
6000    !
```

```
10    SUB Addres(N2,N5)
20    OPTION BASE 1
30    !
40    ! ** PROGRAM TO CALCULATE ADDRESSES OF DIAGONAL ELEMENTS
50    !    IN BANDED MATRIX WHOSE COLUMN HEIGHTS ARE KNOWN
60    !
70    COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
80    DIM Maxa(N2),Mht(N5)
90    ASSIGN #5 TO "A5:T14"
100   ASSIGN #2 TO "A2:T14"
110   READ #5,1
120   MAT READ #5;Mht
130   Nn=Neq+1
140   Maxa(1)=1
150   Maxa(2)=2
160   Mk=0
170   IF Neq=1 THEN L100
180   FOR I=2 TO Neq
190   IF Mht(I)>Mk THEN Mk=Mht(I)
200   Maxa(I+1)=Maxa(I)+Mht(I)+1
210   NEXT I
220   L100: Mk=Mk+1
230   Nwk=Maxa(Neq+1)-Maxa(1)
240   READ #2,1
250   MAT PRINT #2;Maxa
260   SUBEND
270   SUB Assem
280   OPTION BASE 1
290   !
300   ! ** PROGRAM TO CALL ELEMENT SUBROUTINES FOR ASSEMBLAGE
310   !    OF THE STRUCTURE STIFFNESS MATRIX
320   !
330   COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
340   ASSIGN #10 TO "IELMNT:T14"
350   ASSIGN #9 TO "A5:T14"
```

91

```
360 ASSIGN #8 TO "A5:T14"
370 READ #10,1
380 FOR N=1 TO Numeg
390 READ #10;Numest,Npar(1),Npar(2),Npar(3)
400 CALL Element(#10,#9,#8)
410 NEXT N
420 SUBEND
430 SUB Addban(A(*),Maxa(*),S(*),Lm(*),Nd)
440 OPTION BASE 1
450 !
460 ! ** PROGRAM TO ASSEMBLE UPPER TRIANGULAR ELEMENT STIFFNESS INTO
470 !    COMPACTED GLOBAL STIFFNESS
480 !
490 !   S = ELEMENT STIFFNESS ;  A = GLOBAL STIFFNESS
500 !
510 COM Ng,Modex,Npar(3),Humnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Nk,Nlcase
520 Ndi=0
530 FOR I=1 TO Nd
540 Ii=Lm(I)
550 IF Ii<=0 THEN L200
560 IF Ii>0 THEN L100
570 L100: Mi=Maxa(Ii)
580 L200: Ks=I
590 FOR J=1 TO Nd
600 Jj=Lm(J)
610 IF Jj<=0 THEN L220
620 IF Jj>0 THEN L110
630 L110: Ij=Ii-Jj
640 IF Ij<0 THEN L220
650 IF Ij>=0 THEN L210
660 L210: Kk=Mi+Ij
670 Kss=Ks
680 IF J>=I THEN Kss=J+Ndi
690 A(Kk)=A(Kk)+S(Kss)
700 L220: Ks=Ks+Nd-J
```

```
710  NEXT J
720  L200: Ndi=Ndi+Nd-1
730  NEXT I
740  SUBEND
750  SUB Colsol(N3,N4,N2,Nrm,Kkk,#8)
760  OPTION BASE 1
770  !
780  ! ** PROGRAM TO SOLVE STATIC EQUILIBRIUM EQS. USING COMPACTED STORAGE
790  !      AND COLUMN REDUCTION SCHEME
800  !
810  COM Ng,Modex,Hpar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Hlcase
820  DIM A(Nwk),V(Neq),Maxa(Nrm)
830  !
840  ! PERFORM L*D*L(T) FACTORIZATION OF STIFFNESS MATRIX
850  !
860  Nn=Neq
870  IF Kkk-2<0 THEN L40
880  IF Kkk-2>=0 THEN L150
890  L40: ASSIGN #2 TO "A2:T14"
900  ASSIGN #3 TO "A3:T14"
910  READ #2,1
920  MAT READ #2;Maxa
930  READ #3,1
940  MAT READ #3;A
950  FOR N=1 TO Nn
960  Kn=Maxa(N)
970  Kl=Kn+1
980  Ku=Maxa(N+1)-1
990  Kh=Ku-Kl
1000 IF Kh<0 THEN L110
1010 IF Kh=0 THEN L90
1020 IF Kh>0 THEN L50
1030 L50:K=N-Kh
1040 Ic=0
1050 Klt=Ku
```

```
1060 FOR J=1 TO Kh
1070 Ic=Ic+1
1080 Klt=Klt-1
1090 Ki=Maxa(K)
1100 Nd=Maxa(K+1)-Ki-1
1110 IF Nd<=0 THEN L80
1120 IF Nd>0 THEN L60
1130 L60:Kk=MIN(Ic,Nd)
1140 C=0
1150 FOR L=1 TO Kk
1160 C=C+A(Ki+L)*A(Klt+L)
1170 NEXT L
1180 A(Klt)=A(Klt)-C
1190 L80:K=K+1
1200 NEXT J
1210 L90:K=N
1220 B=0
1230 FOR Kk=K1 TO Ku
1240 K=K-1
1250 Ki=Maxa(K)
1260 C=A(Kk)/A(Ki)
1270 B=B+C*A(Kk)
1280 A(Kk)=C
1290 NEXT Kk
1300 A(Kn)=A(Kn)-B
1310 L110:IF A(Kn)<=0 THEN L140
1320 IF A(Kn)>0 THEN L140
1330 L120:PRINT LIN(2),"STOP - STIFFNESS MATRIX NOT POSITIVE DEFINITE"
1340 PRINT "NONPOSITIVE PIVOT FOR EQUATION ";N
1350 PRINT "PIVOT =";A(Kn)
1360 PRINT "SYSTEM IS UNSTABLE AT DEGREE OF FREEDOM NO.";N
1370 STOP
1380 L140:NEXT N
1390 READ #3,1
1400 MAT PRINT #3;A
```

```
1410 SUBEXIT
1420 !
1430 ! REDUCE RIGHT-HAND-SIDE LOAD VECTOR
1440 !
1450 L150:ASSIGN #2 TO "A2:T14"
1460 ASSIGN #3 TO "A3:T14"
1470 ASSIGN #4 TO "A4:T14"
1480 READ #2,1
1490 MAT READ #2;Maxa
1500 READ #3,1
1510 MAT READ #3;A
1520 MAT READ #8;V
1530 FOR N=1 TO Nn
1540 K1=Maxa(N)+1
1550 Ku=Maxa(N+1)-1
1560 IF Ku-K1<0 THEN L180
1570 IF Ku-K1>=0 THEN L160
1580 L160:K=N
1590 C=0
1600 FOR Kk=K1 TO Ku
1610 K=K-1
1620 C=C+A(Kk)*V(K)
1630 NEXT Kk
1640 V(N)=V(N)-C
1650 L180:NEXT N
1660 !
1670 ! BACK SUBSTITUTE
1680 !
1690 FOR N=1 TO Nn
1700 K=Maxa(N)
1710 V(N)=V(N)/A(K)
1720 NEXT N
1730 IF Nn=1 THEN Lxt
1740 GOTO Nxt
1750 Lxt:READ #4,1
```

```
1760 MAT PRINT #4;V
1770 SUBEXIT
1780 Nxt:N=Nn
1790 FOR L=2 TO Nn
1800 K1=Maxa(N)+1
1810 Ku=Maxa(N+1)-1
1820 IF Ku-K1<0 THEN L230
1830 IF Ku-K1>=0 THEN L210
1840 L210:K=N
1850 FOR Kk=K1 TO Ku
1860 K=K-1
1870 V(K)=V(K)-A(Kk)*V(N)
1880 NEXT Kk
1890 L230:N=N-1
1900 NEXT L
1910 READ #4,1
1920 MAT PRINT #4;V
1930 SUBEND
1940 SUB Write
1950 OPTION BASE 1
1960 !
1970 !   ** PROGRAM TO PRINT DISPLACEMENTS
1980 !
1990 COM Ng,Modex,Npar(3),Numnp,Neg,Nwk,Ind,Numeg,Numeat,Maxeat,Midest,Mk,Nlcase
2000 DIM Disp(Neq),Id(3,Numnp),D(3)
2010 !
2020 !  PRINT DISPLACEMENTS
2030 !
2040 PRINT LIN(2),"D I S P L A C E M E N T S",LIN(2)
2050 PRINT "  NODE";SPA(10);"X-DISPLACEMENT    Y-DISPLACEMENT    Z-DISPLACEMENT"
2060 ASSIGN #1 TO "A1:T14"
2070 ASSIGN #4 TO "A4:T14"
2080 READ #1,1
2090 MAT READ #1;Id
2100 READ #4,1
```

96

```
2110 MAT READ #4;Disp
2120 FOR Ii=1 TO Numnp
2130 FOR I=1 TO 3
2140 D(I)=0
2150 NEXT I
2160 FOR I=1 TO 3
2170 Kk=Id(I,Ii)
2180 Ii=I
2190 IF Kk<>0 THEN D(I)=Disp(Kk)
2200 NEXT I
2210 PRINT USING Pri;Ii,D(1),D(2),D(3)
2220 NEXT Ii
2230 Pri:IMAGE 2X,3D,3X,3(4X,MDD.6DE)
2240 SUBEXIT
2250 SUB Stress
2260 OPTION BASE 1
2270 !
2280 !   ** PROGRAM TO CALL THE ELEMENT SUBROUTINE FOR
2290 !      THE CALCULATION OF STRESSES
2300 !
2310 COM Ng,Modex,Npar(3),Numnp,Neq,Nwk,Ind,Numeg,Numest,Maxest,Midest,Mk,Nlcase
2320 !
2330 ! LOOP OVER ALL ELEMENT GROUPS
2340 !
2350 ASSIGN #10 TO "IELMNT:T14"
2360 ASSIGN #9 TO "A5:T14"
2370 ASSIGN #8 TO "A5:T14"
2380 READ #10,1
2390 FOR N=1 TO Numeg
2400 Ng=N
2410 READ #10;Midest,Npar(1),Npar(2),Npar(3)
2420 CALL Elemnt(#10,#9,#8)
2430 NEXT N
2440 SUBEND
2450 !
```

REFERENCES

1. Wilson, E.L., "SAP - A Structural Analysis Program,"
    Report UC SESM 70-20, Dept. of Civil Engineering,
    University of California, Berkeley, 1970.

2. Control Data Corporation, EAC/EASE2 - User Information
    Manual/Theoretical, 1973.

3. Bathe, K.J., and E.L. Wilson, Numerical Methods in
    Finite Element Analysis, Prentice-Hall, Inc., New York,
    1976.

4. Hewlett-Packard Co., Hewlett-Packard System 45 Desktop
    Computer Operating and Programming, 1977.

5. Hewlett-Packard Co., Hewlett-Packard System 45 Desktop
    Computer Mass Storage Techniques, 1977.

6. Hewlett-Packard Co., Hewlett-Packard System 45 Desktop
    Computer Graphics Programming Techniques, 1978.

7. Zienkiewicz, O.C., The Finite Element Method, McGraw-
    Hill Book Co. (UK) Limited, London, 1977.

8. Ralston, A., A First Course in Numerical Analysis,
    McGraw-Hill, 1965.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia   22314

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, California   93940

3. Chairman, Code 69Mx   1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California   93940

4. Professor Gilles Cantin, Code 69Ci   5
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California   93940

5. Ray Mallory   2
   Code 753
   Naval Coastal Systems Center
   Panama City, Florida   32407