

DOCUMENT RESUME

ED 066 875

EM 010 151

TITLE Proceedings of the 1972 Conference on Computers in Undergraduate Curricula.

INSTITUTION Georgia Inst. of Tech., Atlanta.; Southern Regional Education Board, Atlanta, Ga.

SPONS AGENCY National Science Foundation, Washington, D.C.

PUB DATE 72

NOTE 578p.

EDRS PRICE MF-\$0.65 HC-\$19.74

DESCRIPTORS Art Education; Biology; Business Education; Chemistry; *Computer Assisted Instruction; Economics; Education; Engineering; Geography; Interinstitutional Cooperation; Language Instruction; Mathematics; Physics; Social Sciences; Speeches; Statistics; Teacher Education; Undergraduate Study

ABSTRACT

The 83 papers presented at the 1972 Conference on Computers in Undergraduate Curricula are reproduced in this volume. With computer science specifically excluded as an area of interest for the conference, papers fall under the following headings: biology, business, chemistry, economics, education, engineering, geography, languages and art, mathematics, physics, social sciences, statistics, and a general section on faculty training, software exchange, and shoestring facilities. (RH)

ED 066875

Proceedings of the 1972 Conference on Computers in Undergraduate Curricula

June 12, 13, 14, 1972 Atlanta, Georgia
Sponsored by the Southern Regional Education Board
in cooperation with the
Department of Continuing Education, Georgia Institute of Technology

EM 010 15-1

Proceedings of the 1972 Conference on Computers in Undergraduate Curricula

June 12, 13, 14, 1972 Atlanta, Georgia
Sponsored by the Southern Regional Education Board
in cooperation with the
Department of Continuing Education, Georgia Institute of Technology
with support from the
National Science Foundation

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRO-
DUCED EXACTLY AS RECEIVED FROM
THE PERSON OR ORGANIZATION ORIG-
INATING IT. POINTS OF VIEW OR OPIN-
IONS STATED DO NOT NECESSARILY
REPRESENT OFFICIAL OFFICE OF EDU-
CATION POSITION OR POLICY

Library of Congress Catalog Card Number 72-83189

Printed in the United States of America

Distributed by

the Southern Regional Education Board, Atlanta, Georgia 30313

ii

5

3

CONTENTS

	Page
INTERNATIONAL PANELS	
William F. Atchison, University of Maryland "Report on International Panels"	1
BIOLOGY	
SIMULATION OF BIOLOGICAL SYSTEMS, I	
Chairman, Mary Jane Brannon, Huntingdon College	
Thomas L. Isenhour, University of North Carolina John C. Marshall Howard D. Orr, St. Olaf College "The Use of the Computer in an Undergraduate Ecology Course"	3
Stephen B. Kessell, Amherst College "Quantitative Ecology for Undergraduates"	9
Howard C. Howland, Cornell University "Digital and Analog Computing in General Animal Physiology"	17
SIMULATION OF BIOLOGICAL SYSTEMS, II	
Chairman, Mary Jane Brannon, Huntingdon College	
Ernest M. Salter, Cottey Junior College for Women Barry L. Bateman Gerald N. Pitts, University of Southwestern Louisiana "Computerized Ecology Simulation"	33
Davis S. Hinds Mary Ellen Burrows James C. Horton, California State College "Computer Based Inquiry Investigations in Biology"	37
BUSINESS	
BUSINESS AND ACCOUNTING	
Chairman, B. G. Dunn, Fairmont State College	
William F. Bentz, University of Kansas "Computer Assisted Algorithms Learning in Accounting"	41
Thomas L. Guthrie, Indiana University at Fort Wayne "The Business Core Integrator at Indiana University"	46
Elbert B. Greynolds, Jr., Georgia State University "The Time-Sharing Computer and Intermediate Accounting"	61
Harley M. Courtney, University of Texas at Arlington "Remote Time-Sharing for Education in Business Planning and Control"	71
CHEMISTRY	
CAI IN THE CHEMISTRY CLASSROOM	
Chairman, Charles Merideth, Morehouse College	
J. J. Lagowski S. J. Castleberry G. H. Culp, University of Texas at Austin "The Impact of Computer-Based Instructional Methods in General Chemistry"	77
Ronald W. Collins, Eastern Michigan University	89

**"Computer-Aided Classroom Chemistry Instruction Via
Instantaneous Video Projection of Teletype Output"**

W. P. Sliwinski 97
K. J. Johnson, University of Pittsburgh
"Pitt's Interactive Graphics and Computer-Generated
Repeatable Examination Systems"

INSTRUCTIONAL PROGRAMS IN CHEMISTRY

Chairman, Harris Bucas, Jr., Randolph-Macon College

Joyce H. Corrington, Xavier University 103
Lee P. Gary, Jr.
L. Chopin Cusachs, Loyola University
"Interfacing Students and Computing Through Undergraduate
Chemical Research"

Honald D. Crain, University of Kansas 107
"A CAI Program for Aromatic Organic Syntheses Written in
Time-Sharing FORTRAN"

Alfred J. Lata, University of Kansas 113
"An Interactive Time-Sharing Basic Tutorial Program
Sequence in Introductory Electrochemistry"

ECONOMICS

ECONOMIC AND BUSINESS SIMULATION/GAMING

Chairman, Donald Chand, Georgia State University

Donald P. Cole, Drew University 121
J. William Hanlon, Winona State College
"Micromod: Description, Use and Evaluation of a
Microeconomics Computer Game"

James D. J. Holmes, St. Andrews Presbyterian College 125
"New Approaches to Business Simulations"

Ray Billingsley 129
Stanley Wilson, Texas A & M University
"Computer Simulation of Economic Models for Instructional
Usage"

ECONOMICS

Chairman, Martin Solomon, University of Kentucky

Frank J. Bonello 137
William I. Davisson, University of Notre Dame
"Computer Assisted Instruction in Economics at the University
of Notre Dame"

Frank DeFelice, Belmont Abbey College 155
"Integrating Computer Programs in Economics
via Time Sharing Terminals"

Richard A. Stanford, Furman University 161
"Ceteris Paribus Methodology and Computerized Economics-Business
Models"

EDUCATION

EDUCATION-CAI-CMI

Chairman, Sylvia Chapp, Philadelphia Public Schools

Carol A. Cartwright 167
G. Philip Cartwright, Pennsylvania State University
"An Undergraduate Computer-Assisted Instruction Course in the
Early Identification of Handicapped Children"

Roger H. Geeslin, University of Louisville 177
"Preparing Mathematics Teachers to Use the Computer in
Secondary Schools"

Joan Straughan 183
Raymond F. Latta, Western Washington State College
"Computer-Managed Instruction: A Beginning and a Reality at
Western Washington State College"

SUPPORT OF INDIVIDUALIZED INSTRUCTION

Chairman, Henry T. Lippert, USA Medical Field Service School

Harold L. Schoen, Virginia Polytechnic Institute 189
"A Comparison of Types of Feedback to Student Responses in a
CAI Unit"

Arthur Wagner 195
Ronald Bleed, Joliet Junior College
"Using a Computer to Support the Testing Program in
Audio-Tutorial Biology"

Gail M. Nishimoto 199
John M. Horowitz
Ray E. Burger
Richard F. Walters, University of California at Davis
"Initial Development of Individualized Instruction with
Computer Support"

MANAGEMENT OF EXAMINATIONS AND LARGE ENROLLMENTS

Chairman, Henry T. Lippert, USA Medical Field Service School

C. Obert Henderson 207
Mark Hamaer, Washington State University
"Improving Large Enrollment Undergraduate Instruction with
Computer Generated, Repeatable Tests"

Stanley J. Birkin, University of South Florida 217
"An Analysis of the Use and Effectiveness of EXAMINER: A
Computerized Question Bank and Examination Processing System -
in College of Business Courses at the University of South Florida"

R. G. Franke 221
W. D. Dolphin
G. P. Covert, Iowa State University
C. D. Jorgensen, Brigham Young University
"A Computer-Assisted Method for Teaching Large Enrollment
Lecture Sections: The Biology Phase Achievement System (PAS)"

Michael Baldigo, Indiana University School of Business 229
"Operational Aspects of Computer Written and Scored First Year
College Accounting Progress Examinations"

ENGINEERING

ENGINEERING COURSES

Chairman, Rufus P. Walker, Jr., Centenary College of
Louisiana

Tadao Murata 235
Roland Priemer, University of Illinois
"On the Use of a Computer for Motivating Student Projects in
Undergraduate Courses on Network Theory"

Arthur Houghton 241
George H. Quentin
Bruce R. Peterson, University of New Mexico
"A Course in Computer Simulation and Analysis for
Scientists and Engineers"

Donald C. Amoss 249
John N. Gowdy, Clemson University
"Realization and Classroom Application of a Display-Based
System for a Small Digital Computer"

ENGINEERING APPLICATIONS

Chairman, Bernard Waigman, Loyola College

Paul T. Moriarty, New York City Community College 255
"Computer Assisted Numerical Control Part Programming"

W. Waverly Graham, III 265
Don S. Harner, Georgia Institute of Technology
"An On-Line Minicomputer in the Nuclear Engineering Classroom"

Richard Schubert 273
Joseph H. Gill, Western Michigan University
"Computer Applications to Kinematic Synthesis of Four Bar
Mechanisms"

Hinrich R. Martens 277
Stephen G. Margolis, SUNY, Buffalo
"An Analog Computer Optimized for Undergraduate Instruction"

GEOGRAPHY

GEOGRAPHY

Chairman, A. A. J. Hoffman, Texas Christian University

Philip M. Lankford, University of California 285
"Spatial Model Building in the Social Sciences"

Vincent H. Malstrom, Middlebury College 291
"The Computer in Undergraduate Geography at Middlebury"

Paul E. Lovingood, Jr. 295
David J. Coven, University of South Carolina
"The Use of Computers in Geographic Instruction as a Means
for Stimulating Interest in Statistical Methods"

Nancy B. Hultquist, University of Iowa 299
"Introducing Undergraduate Geographers to Quantitative
Analysis Through a Regionalization Framework"

LANGUAGES AND ART

LANGUAGES AND ART

Chairman, Haswell Simpson, Hampden-Sydney College

Anna Marie Thames, Golden West College 305
"CAI and English: A Tentative Relationship"

Robert Phillips, Miami University 311
"Drilling Spanish Verb Forms on Remote Terminals"

Grace C. Hertlein, Chico State College 317
"Computer-Aided Graphics as an Art Form for the Artist"

MATHEMATICS

MATHEMATICS I

Chairman, G. P. Weeg, University of Iowa

Thomas Ralley, The Ohio University 321
"Use of the Computer in Introductory Algebra"

Allen D. Ziebur, State University of New York "A Time-Sharing Computer in the Differential Equations Course"	325
H. M. McAllister, Moravian College "The Role of the Computer in Real Analysis"	329
MATHEMATICS II	
Chairman, Gerald L. Engal, Pennsylvania State University	
Arthur E. Falk Richard Houchard, Western Michigan University "Computerized Help in Finding Logic Proofs"	335
Dominic Soda Aaron H. Konstan Judith Johnson, The Lindenwood Colleges "Computers, Clay and Calculus"	343
Everett Hatner, Hampshire College "APL at Hampshire"	349
Roger B. Kirchner, Carleton College "Computer Generated Pictures for Teaching Calculus"	359
PHYSICS	
QUANTUM PHYSICS	
Chairman, Wayne Lang, MacMurray College	
Carol Bennett, University of Illinois "Computer-Based Education Lessons for Undergraduate Quantum Mechanics"	369
John Merrill, Dartmouth College "Introductory Quantum Mechanics and the Computer"	375
COMPUTER GRAPHICS AND PHYSICS	
Chairman, Arthur Lushmann, Dartmouth College	
David Grillot Jeffrey D. Ballance Larry H. Hubble, Oregon State University Tim. G. Kelley, Southern Oregon College "Interactive Classroom Graphics"	383
Herbert Peckham, Gavilan College "Computer Graphics in Physics"	397
Alfred Bork Richard Ballard, University of California at Irvine "Computer Graphics and Physics Teaching"	409
PHYSICS	
Chairman, Alfred Bork, University of California at Irvine	
Harold Weinstock, Illinois Institute of Technology "Statistical Physics Computer Applications"	417
Charles P. Munch, University of California at Irvine "An Interactive Computer Teaching Dialog for Solving a System of Coupled Oscillators"	427
W. E. Bron, Indiana University "Project CAPLIN: Computer Aided Physics Laboratory Instruction for Non-Science Majors"	433

SOCIAL SCIENCES

HISTORY

- Chairman, Joseph Raben, Queens College
- Betty L. Jehn, University of Dayton 439
"A Computer-Assisted Instruction Program in American History, 1870-1921"
- Charles M. Dollar, Oklahoma State University 449
"A Preliminary Report on Computer-Assisted Learning in American History Courses at Oklahoma State University"

POLITICAL SCIENCE

- Chairman, Calvin Miller, Virginia State College
- William D. Coplin 459
Michael K. O'Leary, Syracuse University
"Educational Uses of PRINCE"
- Bruce D. Bowen 463
Wayne K. Davis, The University of Michigan
"VOTER: A Social Science Data Analysis Program"
- James E. Harf, Ohio State University 467
"A Student Handbook of Empirical Evidence: The Utilization of CAPE Data in Undergraduate Education"

SOCIAL SCIENCE I

- Chairman, Ronald Stiff, Illinois Institute of Technology
- Daniel Vandepoortels 477
Ronald Stiff, Illinois Institute of Technology
"The Creation and Diffusion of Innovative Uses of the Computer in Sociology Education"
- Joseph R. Denk, N. C. Educational Computing Service 483
"POISSON--A Daughter of Dartmouth's IMPRESS Has Been Born in the Environment of IBM Time-Sharing"
- Thomas P. Kershner, Union College 489
"Computer Applications for Social Scientists"

SOCIAL SCIENCE II: SIMULATION

- Chairman, James Hogge, George Peabody College
- Arthur O. Cromer 493
John B. Thurmond, University of Louisville
"Toward the Optimal Use of Computer Simulations in Teaching Scientific Research Strategy"
- John Bartholomew 495
Judith Johnston
Aaron H. Konstam, The Lindenwood Colleges
"A Serious Game as an Introduction to Urban Planning"
- Marshall H. Whitted, Temple University 505
"Political Simulation and the Mini-Computer...A Challenge to the Industry"

STATISTICS

STATISTICS I

- Chairman, William G. Bulgren, University of Kansas
- Elliot A. Tanis, Hope College 513
"Theory of Probability and Statistics Illustrated by the Computer"

Marc S. Weiss, Washington State University "PSYSTAT--A Teaching Aid for Introductory Statistics"	521
Herbert L. Dershen, Hope College "A Course on Computing and Statistics for Social Science Students"	525

STATISTICS II

Chairman, David G. Feinman, Hollins College	
S. C. Wu, California State Polytechnic College "An Alternative Approach in Teaching Statistical Methods"	529
Frederick S. Halley, SUNY, Brockport "Individualized Instruction in Basic Statistics: An Experiment in Computer Managed Instruction"	533
Robert Fletcher Clark I. Williams, Missouri Southern State College "Through Multiple Regression and Threeway ANOV in Sophomore Level Applied Statistics for the Behavioral and Natural Sciences: Instructor-Student Demonstration of the Marchant Cogito 1016 PB+IOTA - 2"	539

GENERAL

FACULTY TRAINING AND SOFTWARE EXCHANGE

Chairman, Judy Edwards, Northwest Regional Education Laboratory	
Joseph R. Deak, North Carolina Educational Computing Service "CONDUIT--A Concrete Pipeline for Software-Starved Little People"	547
Ronald L. Code, Stanford University "An Experiment in Computer Training for College Faculty"	555
L. D. Kugler J. M. Snider, University of Michigan at Flint "Friendly Persuasion: Initiating Reluctant Faculty to the Computer in the Classroom"	567

SHOESTRING FACILITIES

Chairman, Herbert Packham, Gavilan College	
W. Thomas Bass, Mason Junior College "A Computerized Physics Laboratory"	575
Don Leslie Lewis, Bee County College "Measurement of an Automobile's Fuel Consumption, Road Horsepower, Maximum Speed and Maximum Acceleration"	577
John P. Tucciarone, St. John's University "Infinite Sequences and Series Via the Computer"	585

FOREWORD

This proceedings is for the third of a series of conferences on computer uses in undergraduate curricula supported by the National Science Foundation through its Office of Computing Activities. The 1970 Conference was sponsored by the University of Iowa and the 1971 Conference by Dartmouth College. The 1972 conference was planned by a national steering committee consisting of Gerald L. Engel, Pennsylvania State University (on leave from Hampden-Sydney College), John W. Hamblen, Southern Regional Education Board, Glenn R. Ingram, Washington State University, Thomas E. Kurtz, Dartmouth College, Gerard P. Weeg, University of Iowa and Fred W. Weingarten, Claremont Colleges.

Eighty-three(83) papers were selected by panels of referees from the 173 papers sent in for consideration and are reproduced in the following pages. We are indebted to Jerry Engel for organizing and administering the paper selection process and to Glenn Ingram for preparing the "Call for Papers."

Computer Science was specifically excluded as an area of interest for the conference. Papers on computer services are included only if they have novel features or if the services are incidental to other topics.

This document was prepared using the ATS (Administrative Terminal System) with the IBM 360/50 computer in the Information Processing Systems Department of the Atlanta Public Schools. The text was entered into the computer via NOVAR terminals by the Vision Center of Atlanta Public Schools. The splendid cooperation obtained from Thomas McConnell and Marion Boyles, heads of these two departments, respectively, was essential in carrying out this huge task. Particular recognition of Linda Reagan of the Vision Center is due for her constant participation in the typing of the papers and the corrections, to Edward Peabody who did all of the paste-up work associated with the preparation of the camera-ready copy, to Myra Peabody and her friends for the many hours they devoted to proofreading and to Wade Royston, SRES Publications Assistant, for his guidance.

Throughout these proceedings you will find several designs created by the students of Grace J. Hottelien (see her paper p. 317) in her course Computer-Aided Graphics as an Art Form at Chico State College. They afford a pleasant break in the monotony of the printed page and we are grateful for them.

The papers were assumed to be ready for publication as submitted and editing was done only in extreme cases. Every attempt was made not to introduce errors in spelling, typing, etc. during the copy preparation process. In doing so we have discovered and removed some of the authors' preparation errors. We hope that the end result is at least no worse for having passed through our hands - indeed we believe that the net result of our efforts is on the improvement side.

To enhance future communication between the readers and the authors we have included the telephone number of the author when it was available and a concerted effort was made to furnish the zip code. The reward for our efforts will depend upon how you, the reader, and the conference participants are able to benefit from these proceedings, the conference presentations, and what are usually the most valuable - the informal discussions and personal contacts which are afforded by such gatherings of highly motivated persons.

The local arrangements for the 1972 conference were made with the assistance of the Department of Continuing Education, Georgia Institute of Technology, under the direction of Richard Diegand, and its staff, particularly, Bob Herndon and Ken Collins. The chairman was also assisted by a Local Hosts Committee consisting of Euell Evans, Emory University, Thomas McConnell, Atlanta Public Schools, Bob Pearson, University System of Georgia Computer Network, Vladimir Slawicka, Georgia Institute of Technology, Grover Simmons, Atlanta University Center Corporation, and William Wells, Georgia State University. The extensive effort devoted to the planning and operation of the "Computer Fair" by Bob Pearson and his staff deserves special mention as does also Suzanne Bowman, the Chairman's Secretary, for her dedicated attention to communications and registration.

Finally, this series of conferences would not have been possible without the unswerving belief that the computer can and should be used to improve the quality of undergraduate education which is held by Arthur Helmed and Andrew Molnar of the National Science Foundation's Office of Computing Activities, their encouragement and their support. Nor would the conferences be possible without the contributions by the many authors, referees and attendees. To these we express our gratitude and hope that their rewards are many.

John W. Hamblen
1972 Conference Chairman

REPORT ON THE INTERNATIONAL PANELS

William F. Atchison
University of Maryland
College Park, Maryland 20742

A group of twelve speakers from other countries have been invited to take part in panel discussions at this conference. Each will give a report on some aspect of the use of computers in his country, some speaking on applications of computers in various subjects and others speaking more directly on computer science education at various levels. All members of the group are involved in education and the use of computers in secondary schools, and more particularly, all are interested in the training of teachers for secondary schools. At the conference there will be two panel discussions - "Computer Applications in the Sciences Abroad," and "The Status of Computer Education in Other Countries."

Following the conference, the group is going to have a workshop to prepare materials for a new booklet entitled "Aims and Objectives of Computer Studies in General Education" describing a college course for the training of secondary school teachers of computer education. This work will be an extension of the International Federation for Information Processing (IFIP) booklet entitled, "Computer Education for Teachers in Secondary Schools, An Outline Guide," published by the IFIP Working Group on Secondary School Education (WG 3.1) in September 1971. This new booklet will be the first in a series of booklets to be published jointly by WG 3.1 of IFIP and the Organization for Economic Cooperation and Development (OECD). The panel discussions and the work on the new booklet is a joint effort of IFIP and OECD and is sponsored jointly by the National Science Foundation and the U. S. Office of Education. A brief listing of the speakers, along with a comment on their reports and background, is given below.

Alfred Berger from Vienna, Austria is Head of the Austrian School Computer Center. He works on courses for pupils in secondary schools, courses for post secondary vocational training and courses for teachers. The Center also does administrative work for the Ministry of Education including the establishment of data banks of teachers of secondary schools and data banks of pupils in one section of Austria. His panel presentation will be on "The Impact of Computer Science on the Teaching of Mathematics."

Glen Bonham, from Toronto, Canada, works for the Department of Education. He has been a teacher and is now involved in the development of computer science and data processing courses for teachers. He will report on "Computing in the Schools of Ontario, Canada." He will describe secondary school courses and how they relate to University courses and also report on some education research projects.

Uffe Brondum is from Copenhagen, Denmark, where he works in the Offices of the Directorate for Vocational Education. He has helped plan the development of computer education in the Danish educational system. He is in a good position to give the latest news about the state of the art in computer education in his country since he recently lectured on this subject in Denmark. The title of his talk is "A Report on Activities in the Field of Computer Education in Denmark."

E. Garcia Canarigo from Madrid, Spain is on the staff of the Computing Center at the University of Madrid. He teaches courses in informatics and is interested in programming languages and computer education at the secondary level. He has been a speaker on computer education at many international conferences and will report to us on computer education in Spain.

Jacques Hebenstreit is from Paris, France. He teaches informatics at the Ecole Superior d'Electricite and at the University of Paris. One of his main theses is that the ideas of informatics should be embedded in the teaching of many subjects. He has played a key role in the training of large groups of teachers for secondary schools. The title of his talk is "Teacher Training in Informatics in General Secondary Education in France." He was selected to be the program chairman for the Second World Conference on Computer Education to be held in France in 1975.

Andre Kirchberger of Paris, France will chair the panel discussion on "Computer Applications in the Sciences Abroad." He has played an integral role in a number of international seminars and conferences particularly in the area of computer education at the secondary level. He has helped to coordinate the international aspects of this conference. Dr. Kirchberger works for the Centre for Educational Research and Innovation of the Organization for Economic Cooperation and Development (OECD) and is in a unique position to speak not only of developments in France, but also of international cooperation.

R. E. J. Lewis, from Bridges Place, London, United Kingdom, is a lecturer in mathematics at Chelsea College of Science and Technology, a branch of the University of London. He is assisting in the development of new science courses which are based on learning and understanding by discovery. Experiments are simulated on the computer and simulation packages are made available to pupils via interactive terminals. He has a background in physics as well as mathematics and is the Director of an associated computing project. The title of his talk is "The Development of Simulation Packages for the Teaching of Science."

Frank B. Lovis is from Walton, Bletchley, Buckinghamshire United Kingdom. He is on the faculty of The Open University and is developing curriculum materials for computer education. He gave a paper at the Seminar on Computer Sciences in Secondary Education held at Savres, France, March 9-14, 1970, entitled "Teacher Training and Retraining in Computer Sciences." The concepts and innovative methods of The Open University have received much publicity and wide acclaim. Mr. Lovis will give a report based on his own experiences with The Open University.

Timo Halmberg, from Uppsala, Sweden is with the Department of Education and is interested in educational technology as well as computer education. He took part in the CERI/OECD Conference on Computer Sciences in Secondary Education held in Paris, France during June 21-25, 1971. He will report on the developments in computer education in Sweden.

E. D. Tagg is from Bailrigg, Lancaster, United Kingdom. He is a senior lecturer in the Mathematics Department of the University of Lancaster, conducting a course in Computer Oriented Mathematics for Students of Biological and Social Sciences, and chairing the Mathematics panel in the School of Education. He has spent thirty years as a mathematics teacher, having introduced computer studies starting in 1957. He was Chairman of the British Computer Society Schools Committee from 1966 to 1970. The title of his presentation is "Some Varieties of Approach to the Teaching of Computer Studies to Undergraduates."

J. D. Tinsley is from Manchester, United Kingdom and is currently Head of the Schools Project of the National Computer Center. His background is in mathematics, and he has been a teacher for a number of years as well as being Head of the Mathematics Department at St. Edwards School of Oxford. He has been an active member of the British Computer Society Schools Committee as well as an active member of the IFIP Working Group on Secondary Education (WG 3.1). He will report on "The Present Situation in Colleges and Departments of Education in England and Wales concerning the Provision of Courses of Computer Studies for Trainee Teachers."

D. Henk Wolbers is from Voorschoten, Netherlands. He is currently Professor of Information at the Technological University at Delft and has served as Director of the Computer Center at Delft. His original training was as an electrical engineer. He has been very active on international committees in computer science education and is a member of the IFIP Technical Committee for Computer Science Education (TC 3) and the Working Group on Secondary Education (WG 3.1). The title of his talk is "Computers in the University Curriculum in the Netherlands."

The workshop on the preparation of the booklet "Aims and Objectives of Computer Studies in General Education" will also have five representatives from the United States assisting. Each of these is well-qualified to contribute to this effort on the training of secondary school teachers. They are:

William F. Atchison, Chairman of the IFIP Working Group on Secondary Education (WG 3.1) and Director of the Computer Science Center at the University of Maryland.

Sylvia Chapp, a member of WG 3.1 and Director of Instructional Systems for the School District of Philadelphia.

Judy Edwards from the Northwest Regional Educational Laboratory, Portland, Oregon.

David C. Johnson from the Mathematics Education Department of the University of Minnesota.

Thomas Dwyer from the Computer Science Department at the University of Pittsburgh.

THE USE OF THE COMPUTER IN AN UNDERGRADUATE ECOLOGY COURSE

John C. Marshall and Howard D. Orr
Saint Olaf College
Northfield, Minnesota 55057

Thomas L. Isehour
University of North Carolina
Chapel Hill, North Carolina 27514

Introduction

One of the difficulties encountered in teaching an undergraduate ecology course is finding ways to adequately demonstrate and apply the theory. For example, it is possible to construct models of population interactions which demonstrate how a self regulating ecological system such as a forest may develop. But to demonstrate the functional dependencies that are important in such models is somewhat more difficult. Furthermore, the time interval for developmental change or response to perturbation in natural systems is frequently too long to make significant observations to support all or even significant parts of most theoretical models during a one semester course. This is particularly true in northern latitudes where the duration of the field work is limited by the weather. This report concerns our experience with the use of several computer programs to supplement a one semester undergraduate course in ecology. Testing seems to indicate that the use of the computer for gaming in the areas of population dynamics and as an aid to the field analysis of community structure has resulted in a significant enrichment in the ecology course. The programs presently in use are part of a large collection of programs, ranging from very simple to quite complex, and designed to encourage undergraduate biology students to learn to program.

The Course Organization

In lecture the course starts with conceptual examination of eco-systems in terms of overall function in relation to structure. This logically leads to a detailed examination of unit parts, populations, and the isolation and discussion of them. Thus, in general, the flow of the course is initially reductionalistic, leading to a focus on the species populations. At this point the detailed study of population dynamics using computer "experiments" is introduced. Programs illustrating population growth, limited and unlimited, population regulation, competition, predator-prey and a simple ecosystem are made available to the students. These programs allow the student to experiment with parameters and functional relationships and thereby gain a feeling for the nature of the traditional mathematical models of population dynamics. The programs used quite logically introduce the student to a synthetic approach, culminating in a computer model of a simple ecosystem.

Concurrent with the lecture, the laboratory is centered around a study of community structures using natural ecosystems. Here the computer is used to process data from field collections on a day to day basis, encouraging hypothesis testing and making possible rapid modifications in experimental design, while studies of natural communities are in progress.

In summary, two general areas of computer application have proven highly significant in an undergraduate ecology course. First, programs simulating population dynamics have allowed the student to examine for himself the effect of certain population parameters over a period of many generations. These simulation studies lead to an experimentation with a system model, which sharpen concepts in application of theory to management of ecosystems. Secondly, a unified laboratory study of community structure has proven successful using the almost instant data processing of collection data to modify the work as it proceeds. These two points will be discussed in some detail below.

Implementation of the Program

Our experience indicates that generally less than one-fourth of the students who register for undergraduate ecology have had previous computer experience of any kind. This requires that very explicit input directions be given for each of the programs as no attempt is made in the course to teach programming. While the ecological implications and limitations, of each of the models is discussed in lecture, no attempt is made to explain details of coding. All the programs used are written in FORTRAN and are given the status of system programs while the ecology course is in session.

The use of the simulation programs is timed to correspond with and support the discussion of population dynamics in lecture. Special emphasis is given to the effects of key parameters that can be input to the computer models we have. The student is then asked to game, examining,

to his own satisfaction, the effect of varying population regulation parameters. All the programs are designed to accept any number of sets of input and to print out, for each set of input, a record of population density as a function of time or the number of iterations, whichever is appropriate.

The laboratory programs are used primarily to analyze field data, but are presented in such a way as to encourage experimental design prior to data collection.

Programs on Population Dynamics

The computer models used in support of lecture-discussion of population dynamics, listed in their order of presentation to the students are

1. Model of unlimited growth
2. Model of limited growth
3. Model of self-regulation
4. Model of competition
5. Model of predator-prey
6. Model of a simple ecosystem

The first two programs are preliminary in nature respectively utilizing the relationships

$$\frac{dN}{dt} = rN \quad (1)$$

$$\frac{dN}{dt} = rN(1-N/K) \quad (2)$$

where N is the population at time T , r is the reproduction rate constant and (equation 2) K is the carrying capacity of the environment. These two models demonstrate numerically the explosive prediction of equation (1) and the sigmoid prediction of equation (2).

The third model is based on a treatment by Smith[1] and quite adequately presents a population model that includes an input parameter (K), which is indicative of the regulatory pressure on the population. By variation of this parameter the student can demonstrate the following situations:

$K \leq 0$	the population will increase without limit
$0 < K < 1$	population will reach equilibrium without oscillation
$1 < K < 2$	population equilibrium will be approached with oscillations of decreasing magnitude
$K > 2$	the amplitude of the population oscillation will increase without limit

The fourth program presents a model of the competition of two species based on the classical description of Gause[2] i.e. which may be stated as follows:

$$\frac{dN_1}{dt} = r_1 N_1 (K_1 - N_1 - aN_2) / K_1 \quad (3)$$

$$\frac{dN_2}{dt} = r_2 N_2 (K_2 - N_2 - bN_1) / K_2 \quad (4)$$

where a and b are coefficients that respectively state the effect of species 2 on species 1 and the effect of species 1 on species 2. The values of r , K and N are respectively the reproductive rate constant, the carrying capacity of the environment and the initial population for the two species. The students input what they consider appropriate values for the parameters in equations (3) and (4) and the program outputs the population levels for the two populations as a function of time. We have found this to be a very effective way to illustrate clearly this model, which, without illustration is largely concealed by the complexity of the two simultaneous differential equations.

The predator-prey model used is a modification of the treatment presented by Spain[3] and well described by him. The student learns a great deal about predator-prey interaction both from consideration of the numerous factors that must be specified on input and the examination of the graphical output of the program.

Following experience with the five programs above the student is introduced to a program that attempts to implement many of the ideas he has learned into a model of a simple ecosystem with three trophic levels. The required input for each trophic level is as follows:

Plants: initial population, probability of "birth", probability of death, probability of destruction by herbivores, optimum space requirements

Herbivores: initial population, probability of birth, probability of natural death, probability of eating, probability of being eaten, number of time intervals necessary for starvation, optimum space requirements.

Carnivores: initial population, probability of birth, probability of natural death, probability of eating, probability of death by combat, number of time intervals necessary for starvation, optimum space requirements.

The model requires the user to specify the area in which the system exists and this area is assumed to be closed to immigration and emigration. The probabilities specified for each event relate to the arbitrary time scale of the model and are in terms of the event probability for an individual within a given trophic level. Each population event is individually evaluated and the total number of events of a certain type are evaluated as illustrated by the following simple example. Consider an event that has a computed probability (based on the input probability for this event and modified at non-optimum population levels) of 0.5 and a population of two individuals. This means there are three possible outcomes, no event, one event, or two events within a time interval. The probabilities for each of the three possible outcomes can then be derived from the normalized coefficients of a three term binomial expansion and are respectively 0.25, 0.50, 0.25. This argument can be used for a population of any reasonable size. The number of events of a certain type that occur within a given population of N individuals is evaluated by computing the probability that the event will occur to an individual, drawing a random number between 0 and 1 and summing the normalized coefficient of the binomial expansion for N + 1 terms until the summation is equal or greater than the random number. The number of events occurring is then taken as one more than the number of normalized binomial terms summed (the first term is for zero events). A typical output of this program is shown in Table 1. The optimum population levels input for this case are plants, 2500, herbivores, 500, and carnivores, 50. This program has proven useful as a vehicle for the illustration of modelling. The results from the program quite adequately illustrate the nature of a simple ecosystem. Finally, the program is fun. Students have been observed leaning over the printer shouting words of encouragement to one of the trophic levels threatened with extinction.

Laboratory

The laboratory approaches the study of community stability and maturity through comparative studies of the structural aspects of ecosystems. The assumption is made that changes in species diversity within selected trophic levels will reflect decreases in stability and/or maturity due to environmental insults whether natural or man caused.

One ecosystem studied was a local stream in the area of suspected source of pollution. Random samples of benthic invertebrates were collected and categorized at several locations above and below the suspected pollution source. Similar collections were made to compare old fields at different stages of maturity and to measure the extent of injury to a forest which has been selectively logged.

From the collected data species diversity (D)[4] values were computed based on the Shannon-Weaver function from the field of information theory

$$D = -\sum_{i=1}^a p_i \log_2 p_i$$

Where p_i is the fraction of the total number of species belonging to the i th species.

The use of automatic data processing as an integral part of the laboratory has been largely responsible for the change of emphasis to quantitative studies of communities. The freedom from the tedious calculations required when large data collections are analyzed has widened the scope of the studies now possible. Almost instant processing of collected data allows students to continually examine the validity of hypotheses and to revise experiments in progress. This is in sharp contrast to previous procedures where students collected great amounts of data before calculations were made and frequently found from the calculations that the data was worthless. Finally, the availability of programs for statistical analysis has encouraged students to test hypotheses that previously could not have been considered because of the time required for calculations.

MODEL OF A SIMPLE ECOSYSTEM

	PLANTS	HERBIVORERS	CARNIVORES	TIME
1	A = 2253.	B = 966.	C = 97.	10
2	A = 2024.	B = 890.	C = 92.	20
3	A = 1901.	B = 782.	C = 87.	30
4	A = 1787.	B = 659.	C = 85.	40
5	A = 1734.	B = 545.	C = 85.	50
6	A = 1724.	B = 508.	C = 80.	60
7	A = 1726.	B = 527.	C = 76.	70
8	A = 1674.	B = 522.	C = 72.	80
9	A = 1658.	B = 463.	C = 75.	90
10	A = 1665.	B = 384.	C = 74.	100
11	A = 1689.	B = 319.	C = 71.	110
12	A = 1736.	B = 260.	C = 74.	120
13	A = 1821.	B = 228.	C = 76.	130
14	A = 1911.	B = 230.	C = 78.	140
15	A = 1979.	B = 216.	C = 75.	150
16	A = 2082.	B = 230.	C = 67.	160
17	A = 2185.	B = 235.	C = 68.	170
18	A = 2277.	B = 262.	C = 72.	180
19	A = 2359.	B = 285.	C = 73.	190
20	A = 2429.	B = 328.	C = 71.	200
21	A = 2443.	B = 377.	C = 73.	210
22	A = 2430.	B = 493.	C = 70.	220
23	A = 2432.	B = 595.	C = 72.	230
24	A = 2314.	B = 665.	C = 75.	240
25	A = 2194.	B = 682.	C = 74.	250
26	A = 2086.	B = 675.	C = 75.	260
27	A = 1988.	B = 686.	C = 71.	270
28	A = 1908.	B = 614.	C = 68.	280
29	A = 1854.	B = 606.	C = 68.	290
30	A = 1790.	B = 571.	C = 68.	300
31	A = 1731.	B = 555.	C = 66.	310
32	A = 1665.	B = 533.	C = 70.	320
33	A = 1661.	B = 477.	C = 71.	330
34	A = 1653.	B = 412.	C = 71.	340
35	A = 1651.	B = 365.	C = 72.	350
36	A = 1700.	B = 281.	C = 72.	360
37	A = 1766.	B = 250.	C = 73.	370
38	A = 1821.	B = 214.	C = 68.	380
39	A = 1937.	B = 215.	C = 64.	390
40	A = 2034.	B = 221.	C = 61.	400
41	A = 2112.	B = 230.	C = 62.	410
42	A = 2189.	B = 252.	C = 63.	420
43	A = 2236.	B = 294.	C = 69.	430
44	A = 2300.	B = 348.	C = 72.	440
45	A = 2344.	B = 428.	C = 67.	450
46	A = 2306.	B = 561.	C = 64.	460
47	A = 2241.	B = 632.	C = 67.	470
48	A = 2122.	B = 678.	C = 66.	480
49	A = 2027.	B = 694.	C = 64.	490
50	A = 1952.	B = 685.	C = 67.	500

TABLE 1.

Conclusion

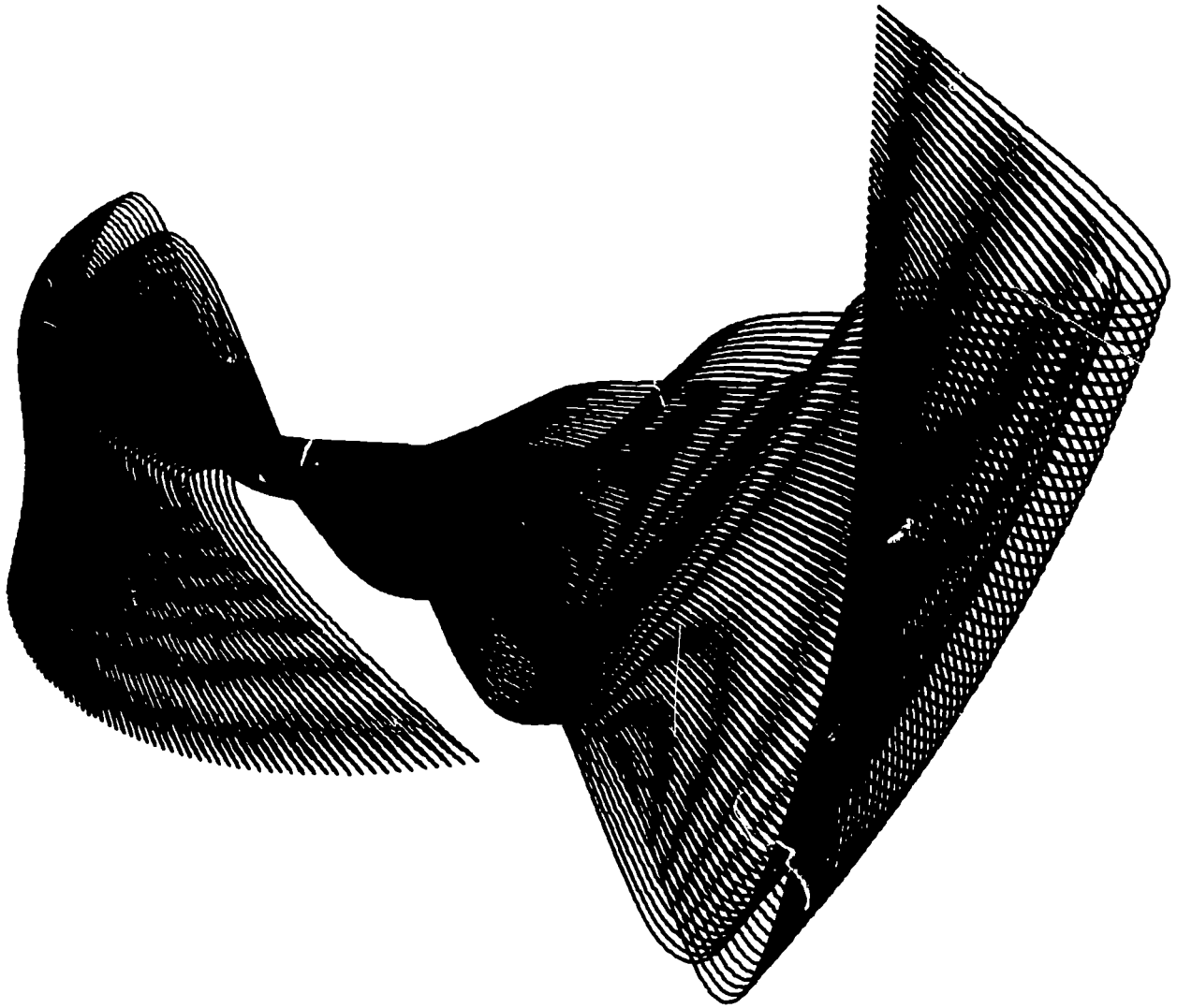
The authors have interpreted the following observations as to mean that computer support of the undergraduate ecology course has been a success:

1. Improvement in test performance in areas relating to population dynamics
2. A dramatic increase in the number of students electing independent study in ecology
3. Computer job log records that indicate that a significant number of students did more than the minimum required amount of gaming
4. Student interest in lab has increased
5. The amount and quality of student data and the interpretation thereof has improved greatly
6. The student morale in lab is noticeably better

We are pleased with the present and planning more extensive computer use in the ecology course in the future.

REFERENCES

1. Smith, J. M., 1968. Mathematical Ideas in Biology, Cambridge.
2. Gause, G. F., 1932. "Ecology of populations." Quart. Review Biol., 7:27-46.
3. Spain, J. D., 1970. Some Computer Programs for Biological Sciences. Mich. Tech. Univ.
4. Shannon, E. E., and Weaver, W., 1963. The Mathematical Theory of Communication. University of Illinois Press, Urbana.



Edwin C. Young

CONTINUUM by Edwin Young

Problem: Develop a graphic using functions, departing from the stereotyped final presentation

QUANTITATIVE ECOLOGY FOR UNDERGRADUATES

Stephen R. Kessell
Amherst College
Amherst, Massachusetts 01002
Telephone: (413) 542-3125

The past decade has seen a dramatic shift in ecology towards detailed quantitative techniques, systems analysis and the mathematical modeling of biotic-abiotic interactions. The holistic approaches of the International Biological Program, the Hubbard Brook Experimental Forest, H. T. Odum's (1971) analog models and Margalef's (1968) applications of cybernetics to ecology have become familiar to undergraduates early in their training. A number of texts popular at the undergraduate level, including Eugene Odum's (1971) Fundamentals of Ecology, E. B. Ford's (1971) Ecological Genetics and several titles in the "Current Concepts in Biology" series (especially Whittaker's (1970) Communities and Ecosystems and Boughey's (1968) Ecology of Populations) are stressing quantitative techniques and methods of modeling and simulation. Yet many undergraduates - including advanced students conducting serious research - lack the background to critically evaluate these ideas and to incorporate mathematical approaches into their own work. This paper will discuss our two years' experiences attempting to alleviate this problem by teaching quantitative ecology at the level of the introductory ecology course at Amherst College.

Ecology at Amherst

Amherst is a small, highly selective men's college firmly devoted to the liberal arts traditions; its student body is outstanding and highly motivated. Within the liberal arts framework, the student has considerable flexibility in constructing his own program, allowing a fairly high degree of specialization in many cases. Although no "ecology" or "environmental science" major is offered, usually from 2 to 5 graduating seniors annually enter graduate school in ecology. Three alternative majors are offered to such students: (1) the regular biology major with specialization and research in ecology and population biology, (2) an interdisciplinary "natural sciences" major including work in biology and two other fields (usually chemistry, geology or mathematics), and (3) a program of "Independent Study," under which all college requirements are waived and the student conducts course work, independent reading and research under the direction of a tutor. Excluding the senior honors seminars in ecology, three formal courses are offered in the field: Biology 41, Ecology; Biology 26, Diversity in Biological Systems; and Biology 40, Aquatic Ecosystems. As all potential ecologists are introduced to the field through the first course, it seemed the best vehicle to introduce quantitative techniques.

Described as "a study of the relationships of plants and animals (including man) to each other and to the total environment," Biology 41 is an elective for sophomores with a prerequisite of genetics; this is occasionally waived for well-prepared or highly motivated students. It is taught by Professor Lincoln Brover, and I take this opportunity to thank him for his complete and extensive cooperation and encouragement in modifying the laboratory portion of his course; I also thank Elizabeth Steele, Academic Coordinator of the Amherst College Computer Center, for her extensive help and cooperation. Scheduled for three hours of lecture and two to five hours of laboratory/field work a week, Biology 41 often requires extra outside lab work, and demands competent and well-written lab reports. As it was decided that new material could not be added to the course that necessitated sacrificing material currently included in it, all changes were made in the laboratory portion of the course.

Computer Facilities

A variety of computational facilities are available to Amherst students. The biology laboratories are equipped with four Wang 360 desk calculators; students have access to several programmable calculators, including a Wang 700 with typewriter output. Four terminals provide APL/360. The academic computer center houses a two-disk, 16K IBM 1130, with high speed reader, line printer and plotter. Although no computer science courses are offered, six non-credit FORTRAN lectures are given each semester, while advanced FORTRAN and APL instruction is available on a tutorial basis. No fees of any kind are charged to students for computer use.

Teaching Quantitative Ecology: Preface

For a number of reasons with which my colleagues may disagree, we feel that the 1130 is our best available tool for teaching programming and mathematical techniques. Although using APL or the desk computers allows a student to concentrate all of his efforts on the program at hand, there is an aura of the "magic box" and a frequent feeling that the machine is doing something mysterious and incomprehensible - something that only "experts" understand. As the student will

certainly be faced with a real system in the near future, we chose to begin with a small system - the 1130 - and teach the rudiments of the monitor and disk operating system while we teach FORTRAN and statistics. The results support our choice.

A number of problems confront any effort to teach quantitative techniques within the framework of an existing course. The majority of the students have never used a computer system or programming language, although perhaps half of the class has had some experience with electronic desk calculators. Their level of proficiency in mathematics includes a semester of calculus at best, and virtually none of the students have any background in statistics. A consideration of the work-load is of paramount importance; it is not possible to merge an ecology course and an introductory statistics/programming course and expect the student to double his commitment for single course credit. If we are not to delete material from an existing course, we must balance the new material to be added against the extra work and time it demands from the students; the single solution in reducing this load is to teach both ecology and mathematics at the same time. The success in carrying out this strategy will in large part determine the success of the course.

Below is a week by week synopsis of the course's laboratory, in which we shall try to evaluate the good and bad points, the successes and the failures.

Phase 1

Biology 41 originally included 10 laboratory exercises; although most have been considerably rewritten, they have all been retained, and serve as the core of our quantitative laboratory.

The student starts using mathematics in the first laboratory. It is a field trip to study light relations in a forest community, and the role of light and abiotic agents as limiting factors. Overlooking the Connecticut River, students compare the forests and understory vegetation of the flood plain to the lower slopes of the Holyoke Range, and in a successional community of eastern hemlocks (*Tsuga canadensis*) measure distances between an individual tree and its nearest neighbor and the mean diameter of the two trees for 100 such pairs of trees. In the report, they are asked to discuss the role of light as a limiting factor in the community based on their own findings and studies in the current literature (Ovington, 1962). A scatter diagram and least squares fit is required, and while doing the fit by hand, it becomes apparent that there must exist an easier way to conduct the repetitive calculations. "Maybe the computer..."

The second lab is the qualitative construction of a food web for a small pond community based on samples collected in the field. While they are writing up the report, we begin teaching quantitative ecology in earnest.

The opening lecture on programming is given in the regular lecture time period (and is the only formal class time sacrificed). It covers a general introduction to the computer as a system - what it is and how it works. Although a number of points are obviously oversimplified, the student learns what a program is, what it means to compile and store a program, what the core storage and disk are, why a user must make certain specifications to the machine, why a programming language understandable to both machine and user is necessary...in short, he has a vague idea of what happens when he pushes the program start button. Finally, the form and format of a FORTRAN program is roughly covered, including the distinction among arithmetic, specification and control statements. A three hour session is scheduled for the following evening. Attendance is not required, but next week's laboratory will require a user-written program to solve a portion of the problem; the report is unacceptable without it.

Perfect attendance... During the first session, each student received the Computer Center's booklet on FORTRAN programming and the 1130 system as a reference/study guide, and the current Newsletter discussing operating procedures (ours is a hands-on system), descriptions and documentation of our general purpose subroutines, special packages and disks available, and the like. These materials are designed to complement the class work, and to serve as readable references (unlike some of the formal user guides). We now pound FORTRAN for two hours.

We start with the control cards and why they are used. We next review arithmetic statements and simple control statements (IF and GO TO); next come READ and WRITE statements. This leads to the use of FORMAT statements (I and F only at this stage). We do not cover vectors and arrays until the next session - after DO loops. In about an hour, we have the basics to allow us to write simple programs.

We have found this to be our best tool - simple programs relevant to the course's lectures and labs. With their notes before them, we pose simple biological problems and write FORTRAN program solutions on the board. We begin with very simple problems (see Figure 1) and involve

OS
21

Figure 1

A Program to Calculate Animal Biomass in a Community

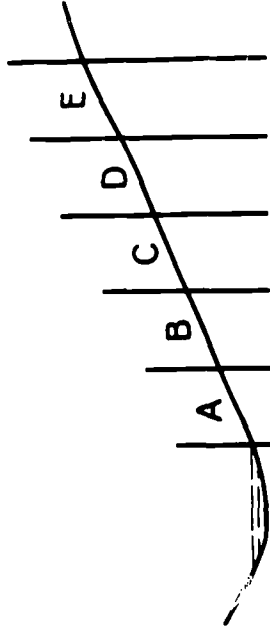
Consider a community with 20 species of animals S_1, S_2, \dots, S_{20} with average biomass B_1, B_2, \dots, B_{20} . We wish to calculate the total biomass of these 20 species.

The FORTRAN program (with control cards):

```
// JOB 0001 1EX0
// FOR
*LIST SOURCE PROGRAM
*LOGS (CARD, 1403 PRINTER)
SUM = 0.
NUMBER = 1
10 READ (2,1) S, B
   FORMAT (2F10.0)
   T = S * B
   SUM = SUM + T
   NUMBER = NUMBER + 1
IF (NUMBER - 20) 10,10,11
11 WRITE (5,2) SUM
   FORMAT (F12.1)
   CALL EXIT
END
// XEQ
      (the 20 data cards)
      1 blank card
```

Figure 2

A Program to Calculate the Centers of Distribution of Vegetation
Consider a transect from river thru flood plain to alone:



with each of 5 zones (A - E) 100 meters wide. Using the river edge as the zero reference, zone A extends from 0 to 100 m. (center at 50 m.), zone B extends from 100 to 200 m. (center at 150 m.), etc.

We consider the distribution of three species - birch - hemlock (H) and spruce (S). The raw data is as follows:

Zone	Number of Individuals		
	Birch	Hemlock	Spruce
A	43	0	0
B	29	12	1
C	17	31	0
D	12	12	28
E	3	7	45

Assume that the distribution within a zone is uniform. We wish to calculate the centers of distribution of the 3 species.

The FORTRAN program:

```
DIMENSION H(5), H(5), S(5)
SUMH=0.
SUMS=0.
SUMS=0.
C=50.
DO 100 I=1,5
  READ (2,1) B(I), H(I), S(I)
  FORMAT (3F10.0)
  SUMH=SUMH + B(I) * C
  SUMS=SUMH + H(I) * C
  SUMS=SUMS + S(I) * C
  C = C + 100.
SUMH = SUMH / (B(1)+B(2)+B(3)+B(4)+B(5))
SUMS = SUMH / (H(1)+H(2)+H(3)+H(4)+H(5))
SUMS = SUMS / (S(1)+S(2)+S(3)+S(4)+S(5))
WRITE (5,2) SUMH, SUMS, SUMS
FORMAT (1E, 'CENTERS OF DISTRIBUTION', 10X, 'BIRCH', 10, 0.5E, 'HDLR',
100X, 'S', 0.5E, 'SPRUCE', 10, 0)
CALL EXIT
END
```

the whole class (of 15 students) in the solution. The response is slow at first, and then picks up. We are certainly aided by the class' small size (which is split into two lab groups), our familiarity with the students and their background, and the use of relevant material from the course. Near the end of the session, we tackle the least squares fit from the first lab and write a program (using one of the available subroutines) that is five statements long and will require five minutes to punch and run. When we look at the output and graphs, most students are thinking of their four hours of tedium and are convinced that the task of learning FORTRAN just might be worth it.

By the end of two hours, everyone has handled cards and output, so we head for the computer center. Within an hour, everyone has learned to use the keypunches and run a program. The third - and last - formal session is scheduled.

The last group meeting continues the writing of simple programs, and introduces the DO loop; through these programs, we meet the need for vectors, arrays, E formats, computed GO TO's, and pauses. We end on the vegetation distribution problem shown in Figure 2, and no one has any real difficulty in writing the FORTRAN program. This program is very similar to the one required in the next lab exercise.

Lab 3 is a study of succession and indicator plants along a transect of a peat bog. The horizontal sequence of species in space compares to the vertical sequence in time; six indicator plant species are counted in seven plots extending from the pond edge to the black spruce (*Picea mariana*) forest. In writing up the report, we make a trade-off; the instructions call for a bar graph of frequencies within the seven plots. This is written for the students and stored on the class' disk; only an XEQ and 7 data cards are required to produce the graph. In exchange, each student is required to solve for the centers of distribution using his own program. A slight curve is thrown - the plots are of unequal sizes, so the center coordinates must be read in as a vector. The author, who serves as laboratory instructor in the course, is available four evenings that week at the computer center. By the time the lab is due the following week, everyone has his program and results. Each student has taken his own problem and field data, written a successful program and produced the needed answers. "It works..." We are now over the hump.

The time required of students and instructor alike is obvious. But a week earlier many of these students had never seen a computer. We (including the students) think it was worth it. The next two labs require no written report and provide both a breather and an opportunity to experiment with the computer. The author is available at the computer center most evenings, while the center personnel are available full-time to assist and teach programming.

Although these two no-report labs are of no mathematical interest, we include a brief description for continuity. Lab 4 is a one hour flight around the Connecticut River valley, taking the pilot, professor and four students at a time. We look at areas already studied, areas about to be studied, the dynamics of a large river system, the geology of the Holyoke Range and the beauty of nature exemplified by New England autumn foliage. Lab 5 follows the same route from the ground - the Connecticut River as a system. We stomp around the flood plain, old channels, oxbows, meanders and successional forests. After all, mathematics is only a tool for handling this fantastic system.

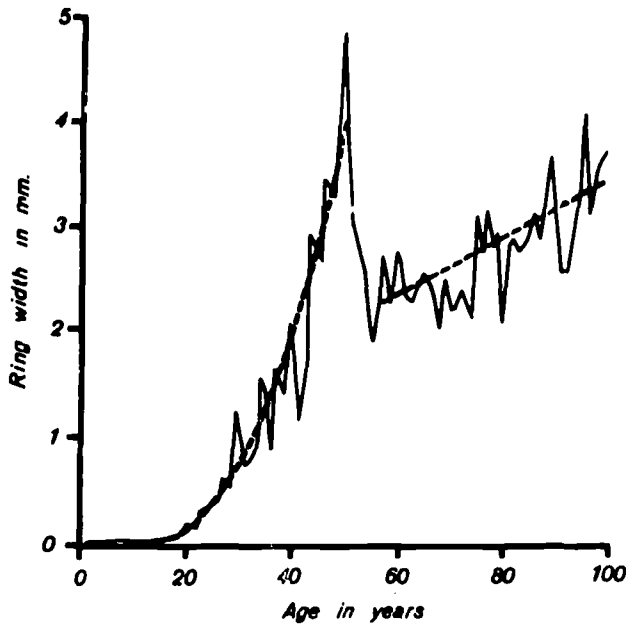
Phase 2

Lab 6 is formally called "Modeling of the physical factors limiting eastern hemlock growth in Pelham, Massachusetts": in one year, it has become known as "The Hemlock Lab" and has earned a certain degree of infamy. To briefly describe the background, eastern hemlock (studied in Lab 1) has an unusual bimodal distribution in southern New England (Kessell and Brover, in press). It is found in the flats and ravines of small streams and on the upper slopes of the lower mountains (below about 2500 feet MSL), but is uncommon on the intervening slopes. In the first lab, we concluded that sunlight and "other factors" limited growth; the students are now asked to not only determine what these factors are, but to quantify their effects as well, and to produce a mathematical model allowing the prediction of annual growth rates from environmental data alone. Furthermore, they are asked to both qualitatively and quantitatively explain the species' bimodal distribution and the environmental and/or genetic basis for it. They have five weeks in which to complete the lab and write it up.

A field trip to the area under study - two stands, one on an exposed hilltop and one at the stream flats at the base of the hill - shows the students the basic differences in flora and the abiotic environment at these extremes where hemlock predominates. But on the intervening slope, under conditions intermediate to those found where the species thrives, hemlock is below the 1% level. Cores are taken from ten individuals at the two stands with an increment borer, and students measure annual rings for 40 years for each tree. They are provided with the general curve of growth rates as a function of age (Figure 3), an equation offering a possible approximation of this function, monthly climatic data for eight variables for the 40 year

Figure 3

Typical growth curve of an individual eastern hemlock

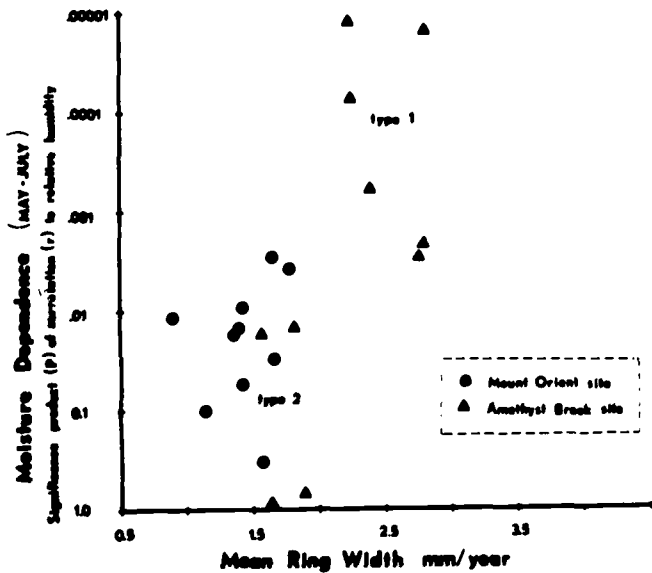


The solid line is the actual annual ring width.
 The dashed line is an approximation of the function relating growth rate to age, which may be written as

$$Y = a + bX + cX^2$$
 where Y is growth rate and X is age
 This equation assumes a constant environment
 The abrupt change to linearity occurs when the tree reaches canopy stature
 The actual growth curve (solid line) is obtained by adding 'environmental indices' F_0 and F_1 , which vary from year to year, to the above equation, giving

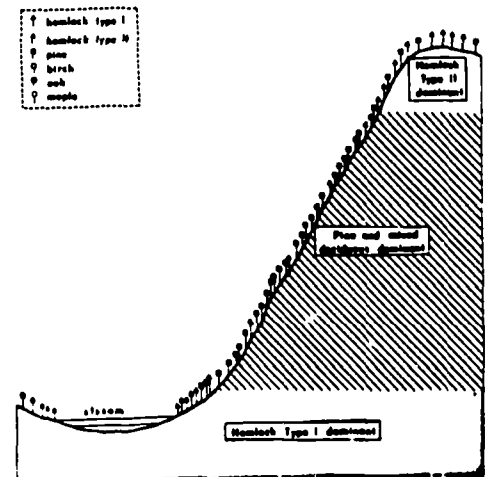
$$Y = a + bX + cX^2 + F_1$$
 (after breaking through the canopy, $F_1 = 0$ and the age function is linear)
 F_0 is used as the dependent variable in multiple regressions of growth on climate; the higher the correlation, the closer the model approximates the actual growth curve
 Correlations with $R = 0.9$ are common (see text)

Growth rates - moisture correlations of *Tsuga canadensis*
 Pelham, Massachusetts



Mean growth rates and correlations to moisture reveal a physiological dimorphism: all individuals from the hilltop and some hemlocks from the valley are the low growth, low moisture type (called Type 2), while some valley hemlocks are the high growth, high moisture morph (Type 1)

Hemlock distribution in southern New England



The dimorphism explains the species unusual distribution.
 High growth Type 1 trees are limited to mesic sites, while drought tolerant Type 2 hemlocks are predominant at more arid sites.
 Apparently neither morph is a good competitor under intermediate conditions. *Tsuga* is excluded by the pines and hardwoods on the intervening sub-mesic slopes

period, a multiple regression program and our best wishes. The author moves to the computer center to offer suggestions, aid and comfort.

It must be pointed out that the current literature does not answer the questions raised in the lab. A number of studies of climatic control of hemlock growth and distribution (including Avery *et. al.*, 1940; Baun, 1950; Olson *et. al.*, 1959; Adams and Loucks, 1971) are furnished to the students, but these findings are somewhat inconclusive and occasionally contradict one another. Studies of both the distribution of hemlock along environmental gradients (Whittaker, 1956) and seedling growth under controlled conditions (Olson *et. al.*, l.c.) suggest ecotypes of the species, but still do not explain the situation the students observe. The student soon realizes that either he is to repeat these earlier attempts, or to find something new not yet in the literature.

Of course, the latter is the case; this problem has been investigated by the author for the past two years, and the lab very closely follows my work - with all the frustration, excitement and dead-ends common to any research problem. And somehow the students become as excited and involved as we are.

A little thought and time in the literature and statistical references (we've found Schaum's Outline in Statistics to be excellent) shows the students that regressions of growth on growing season climate means (using the F_0 component as the dependent variable [see Figure 3]) is a good place to begin. But the correlations are not significant at $P = .05$ (as Avery *et. al.*, l.c., found). Either climate is not a significant factor controlling the annual growth rates, or the growing season means are not adequate. It turns out that the latter is the cause (Pritts, 1966; Kessell and Brower, l.c.); the effects of a factor differ significantly from month to month throughout the growing season. The regressions are now repeated using individual monthly data - a 12-factor regression is completed for each site and each of the eight climate factors. The majority of the multiple R's are highly significant, and the partial r's reveal the monthly changes in climatic effects for each factor. The student is now the proud owner of 385 pages of output.

A week or so with the data gives the basic picture. The environmental gradient from the wet valley to dry hilltop is shown in a number of ways, including the much higher inverse correlations to wind noted at the more exposed site. Temperature not only gives higher correlations at the hilltop site, but an optimum temperature of about 10-13 degrees C. is noted; correlations are direct during months when the temperature is below this range, and inverse when the temperature is above it. In summary, at the hilltop site, direct correlations to moisture and inverse correlations to sunlight during the early spring suggest moisture to be the primary limiting factor. This dependence increases into the summer. At the valley site, light rather than moisture is limiting during the ground saturation of early spring; moisture becomes more and more limiting as the summer progresses. A neat package, with one very bad flaw; the correlations to moisture are higher at the wet valley site than at the more xeric hilltop stand. The original hypothesis that the lower growth rates on the hilltop are due to more severe limiting by moisture is rejected. An alternative explanation is necessary.

At this point, the students have used only the mean growth rates of the ten individuals sampled at each site; they are now provided with the individual growth rates and moisture correlations for every tree. We suggest that they plot mean growth rates vs correlation to moisture (expressed as the product of significances P) for the critical May-July period. The resulting graph (Figure 3) shows the responses clustered into two groups. All individuals sampled at the hilltop and some individuals from the valley exhibit low growth rates and low correlations to moisture, while the remaining individuals from the valley exhibit high growth rates and high correlations to moisture. The differences are highly significant. Ecotypes?...No - both types are sympatric in the valley. Apparently we've found a dimorphism - two specialized morphs physiologically adapted to two different habitats. A model built on these two morphs (with the species excluded on the intervening slope by competitively better-adapted pines and hardwoods) will account for the species' unusual distribution. We also have multiple regression equations (with $R = .9$) giving a mean error of prediction of annual growth rates of less than 5%.

The lab is a long one, but at its completion the students show an immense satisfaction in their work, and hardly gripe about the 20 to 30 page lab reports. They were given a very difficult problem - one in which the mathematics alone would take years by hand. They have answered a difficult question - one which had no answers in the literature. After the reports are handed in, we discuss (with those students who are still interested) other aspects of the problem, including morphological differences, sub-specific hybridization between the morphs, gradient distribution of the two morphs and their roles in interspecific competition (Kessell, in press). There are no readily available answers to their questions; at this point, the students have gone as far as anyone has, know it, and are excited by it. Eight weeks ago they had never seen a FORTRAN program.

Phase 3

For the rest of the course, no more prodding is needed. The students are familiar with the tools and basic statistics and use them as needed. Many are now using APL (at least as a desk calculator) for small jobs, while the desk calculators have become "old hat." The following two labs determine population size, age structure and survivorship curves of sunfish in a small pond; the place of and need for computing tools and methods is obvious. They're on their own, and really don't need our help.

The final two labs study density-dependent survivorship and interspecific competition in Drosophila. Although it is an excellent opportunity to introduce some new computing tools and procedures, time does not permit too much sophistication. The Wang 700 is moved to the lab for 2-way ANOVA's of competing species, and IBM's analog simulator for the 1130 (C.S.M.P.) is made available. So is a little Edmund Scientific analog computer to show basic analog principles. From their data, a variety of methods (each involving different assumption) for determining saturation densities (K) and alpha and beta coefficients of competition are used, and we all argue about the validity of the assumptions.

It strikes me that we've come a long way in ten weeks.

In Retrospect

Perhaps a third of these students will go on in the field to become professionals. To the nonspecialist, we hope to have given both an appreciation of the field and some understanding of current research being carried out. To the future ecologist, we have tried to give a little firmer foothold than he normally would have received. He is not only aware of the tools and materials available, but is eager to incorporate them into his thinking and future work; he is also eager to point out flaws in the thinking of others, including the author's. A small measure of our success has become visible in the students' mathematical sophistication in later courses and independent work. Although obvious time limitations preclude the addition of other interesting material to this course, advanced independent reading and research courses are becoming more common. Within an interterm project, we are now incorporating the tree climate-growth models into two-axis gradient nomograms (following Whittaker, 1967) using IBM's Numeric Surface Techniques package to solve the orthogonal polynomials, and are attempting discriminant analysis of the major components forming the niche hyper-space of Hutchinson (1965) and Whittaker (1967, 1970). At this moment, three sophomores from the 1972 course are using this technique to determine the differential responses of four pine species growing in the same community. Recognizing my own prejudice, these students may well have a three year jump on their contemporaries.

Of course, not all we have done is fun and exciting. We have placed a considerable burden of work on the students. Those truly excited by the field (and not only those with a professional interest) will take the challenge. Those with a lesser interest may feel overwhelmed or inundated with work, and it is critical that the instructors be prepared to recognize this situation and offer help and encouragement when necessary. The distinction between the student having problems he can best work out for himself and the student about to give up is not always obvious. We have to recognize that not all students in our course are on their way to a doctorate in biology, nor are all we excited by the subject matter as we are.

A final word: the instructors of such a course must be prepared for a huge work load. Even with only 15 students, three of us were kept on the go for the entire semester. Professor Brower taught his course, the author taught the quantitative techniques and programming in the laboratory and computer center, and the computer center people kept the center running smoothly. We think it was worth it.

REFERENCES

- Adams, M. S. and O. L. Loucks. 1971. Summer air temperatures as a factor affecting net photosynthesis and distribution of eastern hemlock (Tsuga canadensis L. (Carriere)) in southern Wisconsin. Am. Midland Naturalist 85: 1-10.
- Avery, G. S., H. B. Creighton and C. W. Hock, 1940. Annual rings in hemlocks and their relation to environmental factors. Am. Journal of Botany 27: 825-831.
- Boughey, A. S. 1968. Ecology of Populations. New York: Macmillan. 135 pp.
- Brant, E. L. 1950. Deciduous Forests of Eastern North America. Philadelphia: Blakiston. 596 pp.
- Ford, E. B. 1971. Ecological Genetics, 3rd edition. London: Chapman and Hall. 410 pp.

- Pritts, H. C. 1966. Growth rings in trees: their correlation with climate. *Science* 154: 973-979.
- Hutchinson, G. E. 1965. The Ecological Theater and the Evolutionary Play. New Haven: Yale University Press. 139 pp.
- Kessell, S. R. Ecotypic polymorphism in the eastern hemlock, *Tsuga canadensis*. Submitted to *The American Naturalist*.
- Kessell, S. R. and L. P. Brover, in press. Simulation of the effects of climatic limiting factors. I. Niche variation in the eastern hemlock, *Tsuga canadensis*. *Ecology*.
- Margalef, R. 1968. Perspectives in Ecological Theory. Chicago: University of Chicago Press. 112 pp.
- Odum, E. P. 1971. Fundamentals of Ecology, 3rd edition. Philadelphia: Saunders. 574 pp.
- Odum, H. T. 1971. Environment, Power and Society. New York: Wiley. 331 pp.
- Olson, J. S., F. W. Stearns and H. Nienstaedt. 1959. Eastern hemlock growth cycles and early years. Conn. Agr. Exp. Station Circular No. 205.
- Ovington, J. D. 1962. Quantitative ecology and the woodland ecosystem concept. In J. B. Cragg (ed.) Advances in Ecological Research, Vol. 1, pp. 103-192. New York: Academic Press.
- Spiegel, M. H. 1961. Theory and Problems of Statistics (Schaum's Outline Series). New York: McGraw Hill. 359 pp.
- Whittaker, R. H. 1956. Vegetation of the Great Smoky Mountains. *Ecol. Monog.* 26: 1-80.
- Whittaker, R. H. 1967. Gradient analysis of vegetation. *Biol. Rev.* 42: 207-264.
- Whittaker, R. H. 1970. Communities and Ecosystems. New York: Macmillan. 162 pp.

DIGITAL AND ANALOG COMPUTING IN GENERAL ANIMAL PHYSIOLOGY

Howard C. Howland
Cornell University
Ithaca, New York 14850
Telephone: (607) 256-4716

ABSTRACT

Laboratory and homework exercises in analog and digital computing have been introduced into an upper undergraduate course in general animal physiology in order to increase the amount and depth of presentation of quantitative material. The analog computer was chosen initially for its physical similarity to mathematical flow diagrams of physiological control systems. Subsequently, a small digital computer was added and used both for numerical simulations of control systems and for solution of homework problems requiring only elementary programming skills. A critical factor in the response of students to these innovations appears to be their prior exposure to applied mathematics in the sciences.

Introduction

General animal physiology is a field rich in quantitative topics which deserve, but do not always receive, serious treatment in undergraduate courses. Phenomena surrounding the transmission of the nerve impulse, osmosis, muscle dynamics, enzyme kinetics, countercurrent exchange, and, most generally, homeostatic feedback control systems all require a quantitative presentation if the serious student is to understand them[1].

The mathematical background required for a treatment of these topics is primarily algebra, of which our students generally have adequate command, and differential equations, a knowledge of which our students are usually innocent.

Since a majority of biology students lack a formal background in differential equations, and since the great majority of physiological feedback systems studied contain important nonlinearities rendering them relatively intractable to formal solution, it seemed reasonable to turn to the teaching of analog and numerical techniques for obtaining specific solutions of given feedback systems.

In all candor, I am not now sure that it is possible to do this successfully entirely within the bounds of a physiology course. But one fact was certain when this endeavor began; namely, the problem of providing an adequate mathematical background was not being solved elsewhere in our curriculum.

MATHEMATICAL FLOW DIAGRAMS AND ANALOG COMPUTING

A Flow Diagram Notation

It has probably occurred to many persons that analog computer flow diagrams are almost machine independent, and that they are essentially mathematical structures.

This struck me at a moment when I was particularly vexed with the ambiguous "arrow" diagrams common in physiology. I mean diagrams which purport to say something about the dynamics of feedback loops (often labeled with "EXCITATION" and "INHIBITION") but which, in fact, are the mere mental props of their authors, with little meaning for others.

In any event, I sought to replace these with something specific, and I chose a notation similar to that used in analog computing with a few minor changes in convention (Figure 1)[2]. Several lectures were used to introduce this notation to the students and it was then employed to describe models of physiological feedback loops. (Figure 2.)

One curious fact emerged immediately from my students' attempts to understand these diagrams. On the one hand, the diagrams had an intuitive appeal quite apart from their mathematical precision, as demonstrated by students' ability on tests to answer correctly questions concerning the implications of a diagram, without being able to write the equation system that the diagram represented!

Figure 1.

FLOW DIAGRAM ELEMENTS

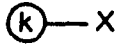
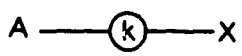
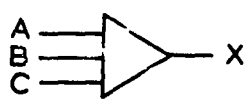
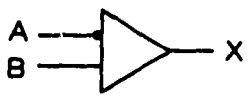
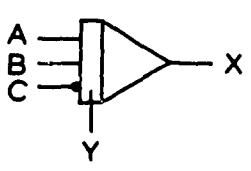
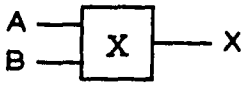
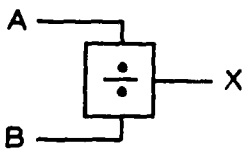

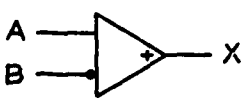
Operation	Symbol	Function
A constant		$X = k$
Multiplication by a constant		$X = kA$
Addition		$X = A + B + C$
Subtraction		$X = B - A$
Integration		$X = Y + \int_0^t (A+B-C) dt$
Multiplication		$X = A \cdot B$
Division		$X = A/B$
Comparison		$X = +1$ if $A > B$ $X = 0$ if $A \leq B$
"Proportional Comparison"		$X = A-B$ if $A-B > 0$ $X = 0$ if $A-B \leq 0$

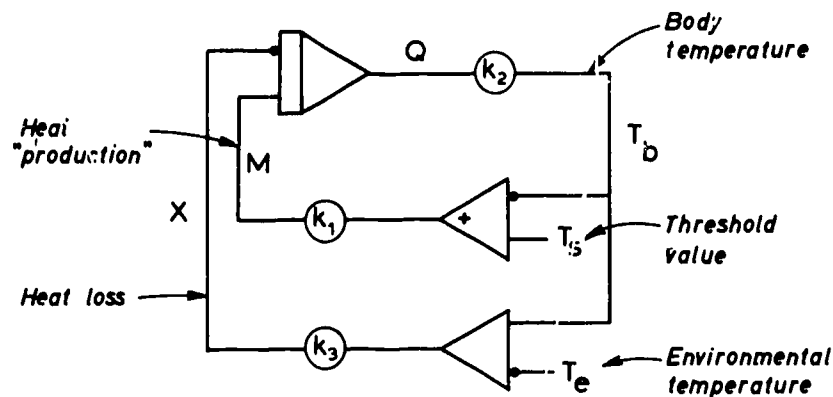
Figure 1 Flow diagram elements. The major difference between this notation and more conventional analog computer notation is the representation of inversion, here a dot at the input. Note that the denominator input in a division operation is denoted by placing it below the numerator input.

Figure 2 A flow diagram for a simple model of temperature regulation. Metabolic heat, M , is "produced" when the body temperature, T_b falls below a set-point, T_s . Heat is generated in proportion to the difference between the set-point temperature T_s and the actual body temperature. Heat is lost from or gained by the body at a rate, I , which is proportional to the difference between the environmental temperature, T_e , and the body temperature T_b . The differential equation for the system is:

$$dQ/dt = k_1 Z + k_3 (k_2 Q - T_e)$$

where $Z = T_s - k_2 Q$ for $T_s > k_2 Q$, else $Z = 0$.

Thermoregulation Model



On the other hand, a single, elementary point continually led students astray. For a long time many students insisted upon seeing the lines of the diagrams as paths through which fluids, hormones, nerve impulses, what have you, - flowed. It was often a hard fight to convince them that the line simply represented a particular variable in the system which could assume a value which might or might not change with time, depending upon the system. (Appeals to their intuition to interpret the lines like voltages or pressures were futile.)

A second pedagogical difficulty arose when I discovered that there was perhaps good reason for the old, imprecise diagrams of the physiology texts--namely, many of the systems which are considered very well understood are, in fact, understood in no dynamic, quantitative way at all. This, of course, is exactly the virtue of a mathematical flow diagram, in that it forces one to record in a systematic way all of the dynamic information that he possesses about a system. However, when one is considering a "classical" physiological feedback system in an undergraduate course, and the quantitative information about it turns out to be precious little, one may well wonder at the wisdom of such a treatment.

The facts are, that while the dynamics of some physiological systems are well understood, others are not, and the physiologist is well advised to choose his examples carefully.

Using the Analog Computer for Lecture Demonstrations

Perhaps the most effective use of the analog computer is in lecture demonstrations. This is because a large number of students can watch the output of the computer simultaneously and the computer can be manipulated by one who is skilled in its operation.

A typical use of the computer in this mode would be to demonstrate a model of the feedback loops involved in the generation of the vertebrate respiratory rhythm. (Figure 3.) The system is of particular interest because one can simulate sectioning parts of the real physiological system by performing the corresponding "operations" on the model system, and show that the behavior of the system is analogous to that of the animal.

Figure 3.

Respiratory Rhythm Model

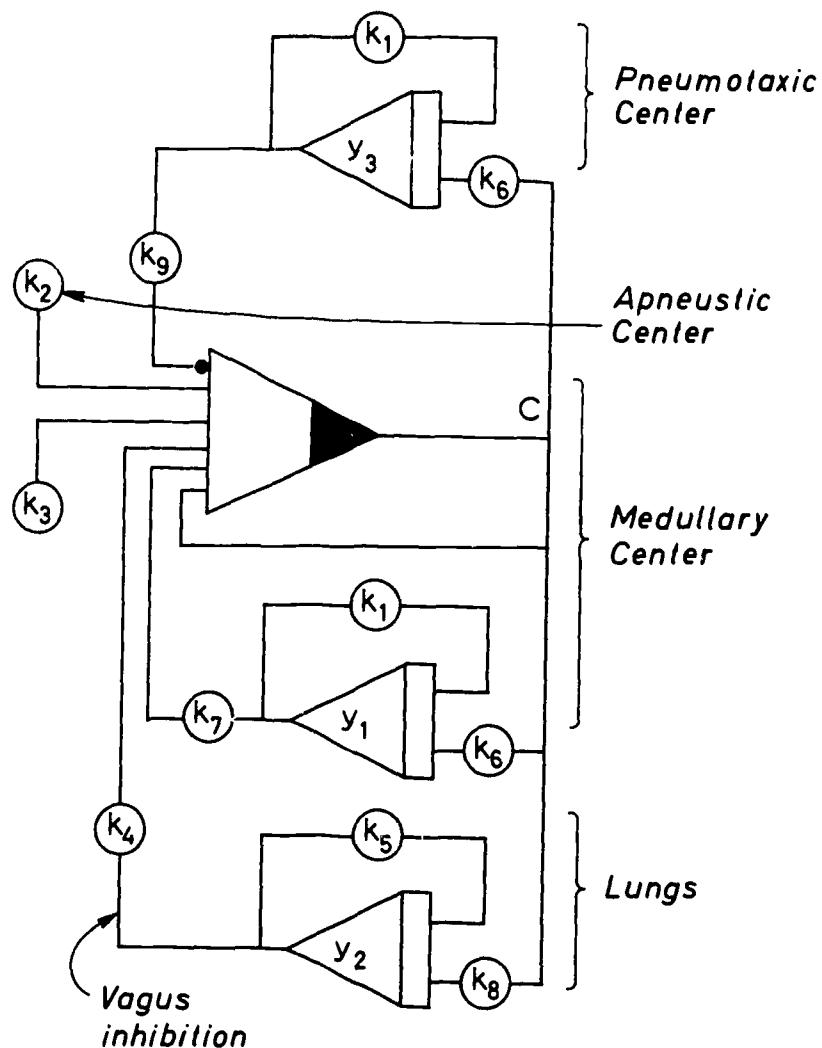


Figure 3 A didactic model of respiratory rhythm generation. The model consists of various qualitative aspects of respiratory rhythm physiology can be demonstrated on the model such as: (a) showing that the rhythm will persist in absence of vagus inhibition but that the respiration will be deeper and slower; (b) persistence of the rhythm in the absence of feedback from the pneumotaxic center in the pons but not without vagus feedback; (c) persistence of the rhythm without the pneumotaxic center and the apneustic center. The model is based on a summary of facts presented in Comroe (1965).

A virtue of the analog computer over the digital computer at the current state of the art should be mentioned here - namely, it provides an economic means for simulating in real time relatively rapid physiological processes which would tax any digital model written in a higher level language.

Analog Computing in the Laboratory

After using the analog computer in lecture demonstrations we were anxious to allow our students in the laboratory portion of the course to have an opportunity to obtain solutions to a simple equation system on it. Since our laboratory students already have had some familiarity with polygraphs and other electronic equipment it is possible to give them a rough idea about the principles on which the analog computer works without their having had an extensive background in electronics.

This year we hope to supplement this background by introducing them to the use of operational amplifiers as part of the signal processing of physiological data. We have constructed batteries of ten operational amplifiers in a convenient plug-board arrangement for this purpose (Figure 4).

In our computing exercises we have students obtain specific solution to a thermoregulatory control loop and to a long term problem in weight regulation which they can then compare to solutions that they obtain on the digital computer (to be discussed below).

A major problem in the use of the analog computer in the laboratory is the limited number of students which can meaningfully work at the computer at one time. The optimum number would appear to be two students at a time and certainly not more than three. This requires a considerable commitment of laboratory and teaching assistant time, as well as a large equipment investment[3].

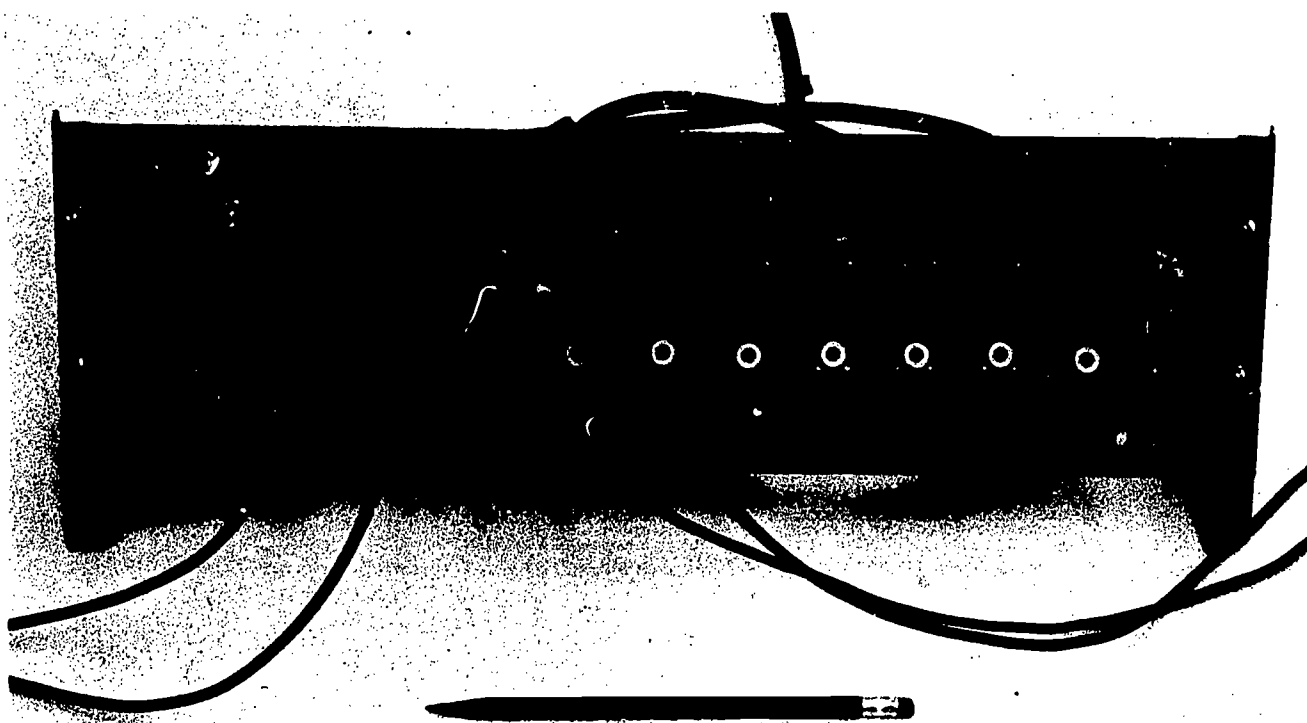


Figure 4 An operational manifold for signal processing in the physiological laboratory. The manifold consists of 10 operational amplifiers and a power supply (Models 105A and 902, Analog Devices Inc.) We have arranged the female plugs in such a fashion that connections between amplifiers can be made with components mounted on standard banana plugs with 3/4" spacing. Each amplifier is provided with two negative input points, two output points, one positive input point, three grounding points and a neutral point, making eight female banana connectors in all. The manifold finds use in adding, subtracting, filtering, and integrating physiological wave forms.

DIGITAL COMPUTING

Introduction

Instruction in digital computing is undergoing a rapid change primarily due to the introduction of mini computers into many curricula. In our own course we purchased a small digital computer in place of a multiple capacitor delay unit for our analog computer. We rapidly discovered that the digital computer offered enormous educational possibilities as a stand-alone computer itself, in many ways more powerful than the (more expensive) analog computer.

Previously, our only access to digital computing had been through batch-processed languages available to students only with very long turnaround times. The mini computer afforded us immediate turnaround and meaningful results in the laboratory via its interactive language, FOCAL. Currently there are a wide variety of mini computers available, all of which offer interactive FOCAL or BASIC, or both, and hence are suitable for the applications discussed below.

"Straight Line" Programming to Solve Quantitative Homework Problems

One of the first uses we put our computer to was to ease the computational load of our students in solving quantitative homework problems. This was done by teaching them "straight line" programming i.e. programming which involved no iteration or logical branches. A student can learn these elementary aspects of programming within about a half an hour at an interactive terminal. This amount of computing knowledge greatly increases his or her computational skills[4].

A typical "straight line" programming problem is given in Figure 5.

C-FOCAL, 1969

```
01.10 ASK ?R?, "PRESSURE IN MM HG ",?P?;!
01.20 ASK ?L,VI?;!
01.30 SET PI=3.14159
01.40 SET P = P*1333.22
01.50 SET Q=PI*P*R+4/(8*L*VI)
01.60 TYPE "FLOW IN CC/SEC", %12.03, Q;!
*GO
R:.08 PRESSURE IN MM HG P:15
L,:4 VI:.04
FLOW IN CC/SEC          2.011
*
```

Figure 5--A "straight line" program in FOCAL (i.e., a program with no logical branches or iterative loops). The program computes the flow of blood, Q, through a hypodermic needle whose length and radius are specified. The pressure drop along the needle and the viscosity of blood are given as input parameters. Use of such a program completely documents the procedures used in the exercise. Readers unfamiliar with FOCAL may translate the program into BASIC by substituting LET for SET, INPUT for ASK and PRINT for TYPE. The program is taken from Howland, (1971).

In learning to write such a program a student has:

- a. obtained command of certain functions (like eEE) which he may not have had before;
- b. learned to formulate the problem in sufficient generality to solve any similar case with different input values;
- c. and (assuming the solution is correct) made a legible record of his method of solution which he can refer to any time in the future or, (if the solution were incorrect), made a record of his method which can easily be corrected in the future.

I believe that the combination of generality and documentation in such elementary programming skills puts our students well ahead of even accomplished students who operate with pencil, paper, and a slide rule. One great advantage of the digital computer is its enforced documentation. This is an enormous boon to a teacher who must either wade through piles of scribbled homework, or take the answers on faith and abandon all hope of correcting his students' mistakes[5].

Numerical Simulation of Physiological Systems

With the introduction of a digital computer into the laboratory we began using it along with the analog computer to simulate physiological systems.

To accomplish this we wrote a general purpose, second order Runge-Kutta routine for solving sets of first order differential equations. The students were then required only to program the differential equations that the Runge-Kutta routine would solve.

Figure 6 shows the listing of the Runge-Kutta routine with equation subroutine and typical printout. It will be noted that the equation system we have simulated for this example is identical to that given for the analog computer respiratory rhythm model in Figure 3.

C-FOCAL,1969

```

01.05 E
01.10 T !!, "ENTER RUN PARAMETERS",!!
01.15 A ?ND, IP, PN, TS, TQ ?,!
01.20 C PLACE INITIALIZATION STATEMENTS HERE
01.22 T !! "SET CONSTANTS",!!
01.25 F J = 1,9;T "K", %3,J," " A K(J);T %5.03, K(J),!
01.30 T !! "INITIALIZE INTEGRATORS",!!
01.50 F J=1,ND; T %3,J;A ?Y(J) ?,!

02.23 S H=(TQ-TS)/(IP*PN)
02.24 S TM=TS
02.30 T !! " TIME Y(1) Y(2) Y(3) PLJT OF Y(2) ",!!
02.31 D 3.35
02.32 F IT=1,PN; D 3.00
02.40 T !!, %6.05,%H ?,%4," NO. OF INTERVALS ",IP*PN;Q

03.33 F I=1,IP;S TM=H*((IT-1)*IP+1)+TS; D 4.0
03.35 T %6.03,!,TM;F I=1,ND;T " ",Y(1)
03.36 D 6.0

04.12 F L=1,ND;S OL(L)=Y(L);S R(L)=0
04.13 F L=1,2;D 8.0;D 9.0
04.14 F L=1,ND;S Y(L)=OL(L)+R(L)/2

06.10 F L=0,.1,Y(2); T " "
06.20 T "*"

08.05 C PLACE DE EQUATION SYSTEM HERE
08.10 S Z=C+K(2)+K(3)-(K(7)*Y(1)+K(4)*Y(2)+K(9)*Y(3))
08.15 S C=.5*FSGN(Z)+.5
08.20 S D(1)=C*K(6)-K(1)*Y(1)
08.30 S D(2)=C*K(8)-K(5)*Y(2)
08.40 S D(3)=D(1)

09.11 F M=1,ND;S R(M)=H*D(M)+R(M)
09.20 I (L-1)*9.3;F M=1,ND;S Y(M)=Y(M)+R(M)
09.30 R
*
```

Figure 6A--A second order Runge-Kutta program for systems of first order differential equations. The program is written in FOCAL. The equations for the differential equation system are in section 8.0, in this case, the equations are for the respiratory rhythm model of Figure 3. The program is sufficiently general that it can be adapted to any equation system by minor changes in statements 1.25 (setting the number of constants), and 2.3 (output titles) and 6.1 (plotter parameters).

ENTER RUN PARAMETERS

ND,3 IP,2 PN,20 TS,0 TQ 20

SET CONSTANTS

K 1 .077 0.077
 K 2 .2 0.200
 K 3 .2 0.200
 K 4 .5 0.500
 K 5 .2 0.200
 K 6 .2 0.200
 K 7 .5 0.500
 K 8 .4 0.400
 K 9 .5 0.500

INITIALIZE INTEGRATORS

1Y(J) 0
 2Y(J) 0
 3Y(J) 0

TIME	Y(1)	Y(2)	Y(3)	PLOT OF Y(2)
0.000	0.000	0.000	0.000	
1.000	0.193	0.362	0.193	*
2.000	0.371	0.658	0.371	* *
3.000	0.536	0.901	0.536	* *
4.000	0.688	1.100	0.688	* *
5.000	0.780	1.163	0.780	* *
6.000	0.722	0.953	0.722	* *
7.000	0.669	0.780	0.669	* *
8.000	0.619	0.639	0.619	* *
9.000	0.573	0.523	0.573	* *
10.000	0.531	0.429	0.531	* *
11.000	0.491	0.351	0.491	* *
12.000	0.455	0.288	0.455	* *
13.000	0.421	0.236	0.421	* *
14.000	0.390	0.193	0.390	* *
15.000	0.361	0.158	0.361	* *
16.000	0.384	0.229	0.384	* *
17.000	0.548	0.550	0.548	* *
18.000	0.700	0.812	0.700	* *
19.000	0.841	1.027	0.841	* *
20.000	0.825	0.923	0.825	* *

H 0.50000 NO. OF INTERVALS 40*

Figure 6B-A "run" of the model with all parts intact. The lung volume Y(2) is also plotted as well as printed. The run parameters are: ND = number of derivatives, IP = number of points computed between each printed point, PN = number of points printed, TS = starting time, TQ = quitting time. H is the step size in time units.

*G 1.22

SET CONSTANTS

K 1 0.077
K 2 0.200
K 3 0.200
K 4 0 0.000
K 5 0.200
K 6 0.200
K 7 0.500
K 8 0.400
K 9 0.500

INITIALIZE INTEGRATORS

1Y(J) 0
2Y(J) 0
3Y(J) 0

TIME	Y(1)	Y(2)	Y(3)	PLOT OF Y(2)
0.000	0.000	0.000	0.000	
1.000	0.193	0.362	0.193	*
2.000	0.371	0.658	0.371	* *
3.000	0.536	0.901	0.536	* *
4.000	0.688	1.100	0.688	* *
5.000	0.830	1.263	0.830	* *
6.000	0.961	1.396	0.961	* *
7.000	1.082	1.506	1.082	* *
8.000	1.194	1.595	1.194	* *
9.000	1.298	1.668	1.298	* *
10.000	1.395	1.728	1.395	* *
11.000	1.338	1.497	1.338	* *
12.000	1.238	1.226	1.238	* *
13.000	1.147	1.004	1.147	* *
14.000	1.062	0.823	1.062	* *
15.000	0.983	0.674	0.983	* *
16.000	0.910	0.552	0.910	* *
17.000	0.843	0.452	0.843	* *
18.000	0.780	0.370	0.780	* *
19.000	0.723	0.303	0.723	* *
20.000	0.669	0.248	0.669	* *

H 0.50000 NO. OF INTERVALS 40*

Figure 6C--A run of the respiratory rhythm model with the vagus inhibition eliminated K.[4] as been set to zero). Note the deeper inspiration and longer period.

In general, one can translate directly from a mathematical flow diagram like that of Figure 3 into the equations of section 8.0 of the second order Runge-Kutta routine of Figure 6A. Our method is to handle the miscellaneous algebraic computations first, and then to write equations for each of the integrator inputs in turn[6].

Since FOCAL is an interactive language and space in the mini computer is at a premium we have not tried to write a completely general program (certain aspects of it, like the plotter in section 6.0, Figure 6A, are changed to fit the needs of the particular simulation at hand). We did try to arrange the program so that in each run all of the important parameters of the run are displayed. Again, this is a decided advantage over the analog computer where a great deal of documentation is left to the user.

On the other hand, as noted above, digital simulation in a higher level language is a slow process; a run of 20 points as in Figure 6B or 6C would take approximately 2.5 minutes and cost 21 cents at our Divisional computing facility.

The example of Figure 6 is a qualitative one, designed to illustrate the redundant feedback loops which must exist in the central nervous system to generate the respiratory rhythms. The point of digital simulation in this case is to allow students to obtain a "feel" for the system which they could not otherwise get.

A more complete use of the digital computer is made in the simulation of a system which is quantitatively well described. Such an example is the respiratory model of Grodins, et al. (1954) cited by Milhorn (1966).

The analog computer program given by Milhorn shows eight operational amplifiers, two integrators and a multiplier. We have modified his diagram into our notation, removing all of the hardware aspects of the analog computer diagram and also scaling constants. (Figure 7.)

This model has been programmed as a subsection of the Runge-Kutta routine and this section together with a run is shown in Figure 8. The equations are direct translations of Milhorn's equations (5-10 through 5-12, page 74 of Milhorn, 1966). Our output may be compared directly with that given in Figure 5-42 of Milhorn's treatment. It might be noted that in this simulation we used a smaller step than previously, computing 10 points for every one plotted.

Table 1

Constants and Variables of the Two Compartment Respiratory Chemostat*
(After Grodins, et al. (1954))

Variable or Constant	Initial Condition or Value	Physiological Meaning
Y(1)	.052	Alveolar CO ₂ concentration
Y(2)	.533	Tissue CO ₂ concentration
K(1)	.00425	Slope of CO ₂ dissociation curve (mmHg ⁻¹)
K(2)	.32	Intercept of CO ₂ dissociation curve
K(3)	3.0	Alveolar compartment volume (liters)
K(4)	.01	Inspired CO ₂ concentration
K(5)	760.0	Barometric pressure (mmHg)
K(6)	40	Lumped tissue compartment volume (liters)
K(7)	.263	Metabolic CO ₂ production (liters/minute)
K(8)	471	Slope of VR vs Y(2) curve (liters/minute)
K(9)	246	Intercept of VR vs Y(2) curve (liters/minute)
VR	5.023	Alveolar ventilation rate (liters/minute)
D(1)	0	Time derivative of alveolar CO ₂ concentration
D(2)	0	Time derivative of tissue CO ₂ concentration
Z	0	Non-dimensional normalized response of VR

*The equation system for this model is given in the flow diagram of Figure 7 or in the program equations 8.05-8.30 of Figure 8.

Two Compartment Respiratory Chemostat (After Grodins et al.)

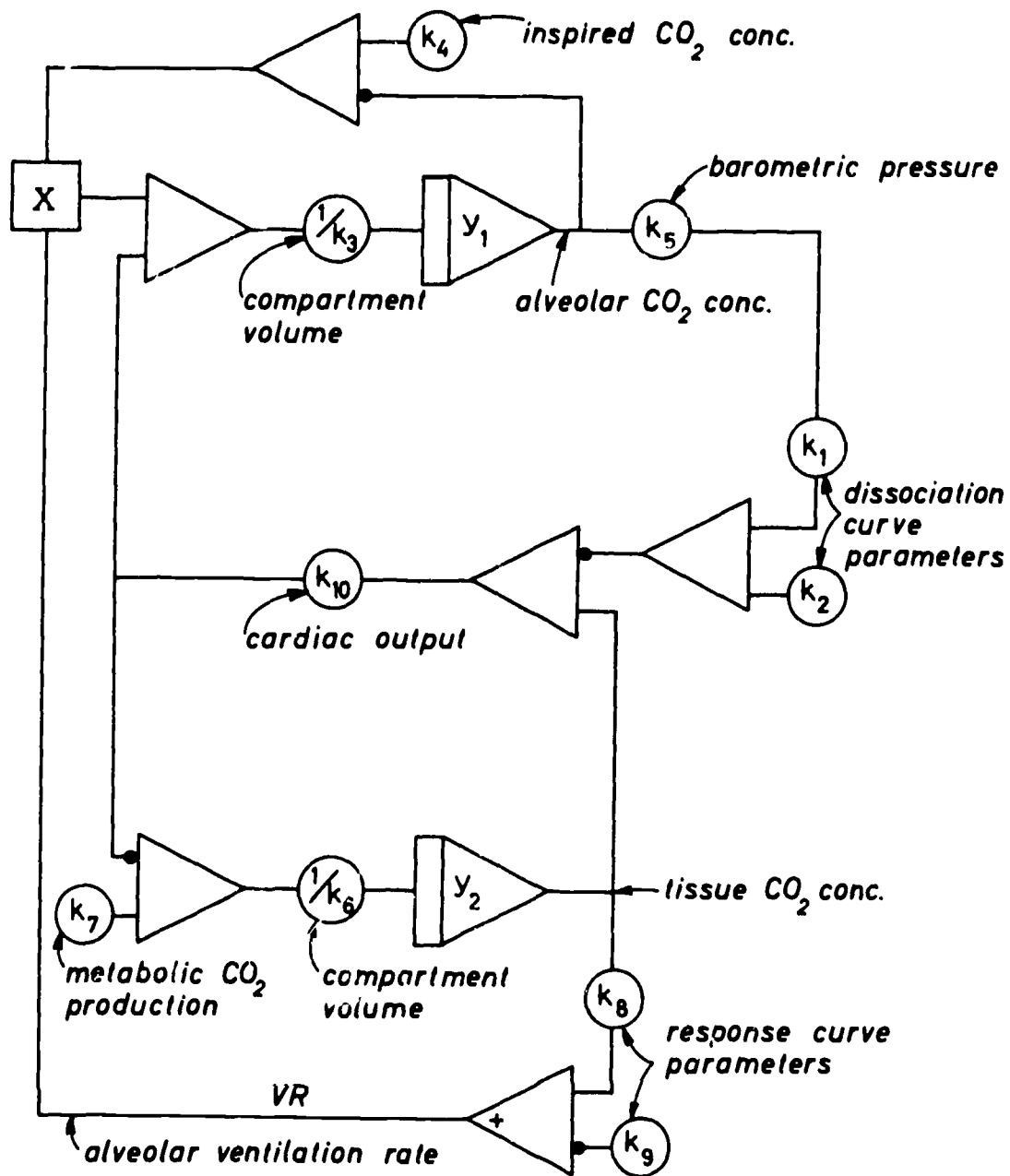


Figure 7 Two compartment respiratory chemostat. The model has been formulated as equations given in Milhora (1966) after a model of Grodins et al. (1954). Normal values of the variables and constants are given in Table I. The model permits investigation of the importance of large number of parameters influencing alveolar ventilation rate.

```

08.05 S VR=K(8)*Y(2)-K(9)
08.10 S D(1)=(VR*(K(4)-Y(1))+K(10)*(Y(2)-(K(1)*K(5)*Y(1)+K(2))))/K(3)
08.20 S D(2)=(K(7)+K(10)*(K(1)*K(5)*Y(1)+K(2)-Y(2)))/K(6)
08.30 S Z=(VR-Z0)/(Z1-Z0)
*
```

ENTER RUN PARAMETERS

```
ND,2 IP,10 PV,20 TS,0 TQ 20
```

SET CONSTANTS

```

Z0, 5.023 Z1 6.106
K 1 4.25E-3 0.004
K 2 .32 0.320
K 3 3 3.000
K 4 0.01 0.010
K 5 760 760.00
K 6 40 40.000
K 7 .263 0.263
K 8 471 471.00
K 9 246 246.00
K 10 6 6.000
```

INITIALIZE INTEGRATORS

```

1Y(J) .052
2Y(J) .533
```

TIME	Y(1)	Y(2)	VR	PLOT OF Z
0.000	0.052	0.533		
1.000	0.054	0.534	5.379	*
2.000	0.054	0.534	5.640	* *
3.000	0.054	0.535	5.806	* * *
4.000	0.053	0.535	5.912	* * * *
5.000	0.053	0.535	5.981	* * * * *
6.000	0.053	0.535	6.025	* * * * *
7.000	0.053	0.535	6.053	* * * * *
8.000	0.053	0.535	6.072	* * * * *
9.000	0.053	0.535	6.084	* * * * *
10.000	0.053	0.535	6.091	* * * * *
11.000	0.053	0.535	6.096	* * * * *
12.000	0.053	0.535	6.100	* * * * *
13.000	0.053	0.535	6.102	* * * * *
14.000	0.053	0.535	6.103	* * * * *
15.000	0.053	0.535	6.104	* * * * *
16.000	0.053	0.535	6.104	* * * * *
17.000	0.053	0.535	6.105	* * * * *
18.000	0.053	0.535	6.105	* * * * *
19.000	0.053	0.535	6.105	* * * * *
20.000	0.053	0.535	6.105	* * * * *

H 0.10000 NO. OF INTERVALS 200*

Figure 8 Runge-Kutta FOCAL implementation of the respiratory chemostat model of Figure 7. The example shows the model's response to a step from 0 to 1% CO₂ concentration of inspired air. The ventilation rate rises in response to the step input, approaching close to its steady-state value within 5 or 6 minutes. The program as given was run on a 4k word computer. Ten points were computed for each point printed.

Such a model can be used in the laboratory to study a variety of aspects of the control system. Answers to all of the following questions may be obtained from running the model with the appropriate values of input constants:

1. How will the rate of alveolar ventilation be affected by:
 - a. increased metabolic CO_2 production?
 - b. decreased cardiac output?
 - c. decreased barometric pressure?
 - d. increased inspired CO_2 concentration?
 - e. increased gain in the alveolar ventilation feedback loop?

Perhaps it is well to point out that these are indeed theoretical questions, i.e. questions concerning a model which may or may not accurately mirror the behavior of the animal. (It would of course be foolish to be carried away by the simulation into thinking that manipulating such a model was a substitute for animal experimentation.)

DISCUSSION

Difficulties in Prerequisites

One of the major problems I have encountered in attempting to introduce a quantitative systems approach into my physiology course has been a feeling on the part of many students that they are not really qualified to cope with physical and mathematical concepts that the course emphasizes. This is in spite of the fact that all the students are required to take courses in physics and mathematics before registering for general animal physiology.

It is not uncommon for me to find, however, that their physics course has omitted a treatment of one of the classical topics which we need for physiology. Two areas in particular which seem to be getting short shrift in some physics courses these days are hydrodynamics and geometrical optics!

On the mathematical side I frequently encounter the following problem: namely, while students are taught in a one year calculus course to differentiate and integrate analytic functions, they do not even know how to write differential equations, let alone solve them by formal methods. I would, of course, be happy if my students could simply formulate the equations, because I can show them how to obtain numerical solutions to them. But the problem is generally at the initial step of writing equations.

In the face of these prerequisite difficulties, there is strong pressure to limit the discussion to qualitative material that is more easily grasped, and, of course, some compromise must be made if the students are to benefit from the presentation.

The problem is accentuated by the fact that there are some students, engineers and mathematicians interested in physiological systems, who do follow very well the quantitative approach and are anxious to refine the models to a point well beyond what their classmates can understand[7].

Our past response to these difficulties has been to attempt to make the mathematics easier and to teach it along with the physiology. We have clearly reached a saturation point however, and more recently we have attempted to shift some of the burden onto other courses.

We now have a one credit course in interactive computer programming within the Division of Biological Sciences which is in its second year. We will shortly be getting students into the physiology lab who are not only trained in using the digital computer, but enthusiastic about its use.

At the same time the Division of Biological Sciences has been instrumental in planning a course in mathematics for biologists which emphasizes equation systems and mathematical models. Students of this course will presumably be more receptive to a systems approach in physiology when they start coming through.

What About Physics?

The problem with the classical physics prerequisites is a more difficult one to handle. (Perhaps this would not be true at another institution.) My own belief is that we must generate some good self-study material in bite-sized packages to supplement our students' curriculum. On the one hand it is impossible to take the time within a physiology course to teach the fundamentals of, say, geometric optics, and even if one had the time, most of the students would

not hold still for it. But on the other hand, it is pointless to try to teach a student about an eye if he doesn't even know how a lens works. So there must be some way that a student can fill these lacunae in his background rapidly at a time when he is motivated to do so. Hence my belief that individual self-study material affords a solution.

Naturally, teachers at different institutions may find a different pattern of deficiencies in prerequisites and different constraints on solutions which attempt to remedy them.

SUMMARY

I have attempted to show how we have used analog and digital computation in teaching quantitative systems analysis in general animal physiology. That approach consists of (1) the employment of a mathematical systems notation similar to that commonly used in analog computation; (2) the physical realizations of such systems on analog computing equipment, supplemented by the use of feedback amplifiers in physiological exercises as signal processing equipment; (3) the use of an interactive digital computer in the solution of quantitative homework problems in elementary programs with no iteration or logical branching; (4) the use of the digital computer to obtain numerical solutions to systems of differential equations.

I have discussed the difficulties of such a course which primarily concern the disparity between the prerequisite knowledge needed and possessed by its students, and I have described some steps which we have taken and are taking to alleviate these difficulties.

ACKNOWLEDGEMENT

Portions of the equipment used in the work reported here were supported by Grant GY 6505 to Cornell University under the Instructional Scientific Equipment Program.

NOTES

1. Peter Steward (Stewart, 1970) has eloquently stated the value of quantitative models in the physiology curriculum. I particularly endorse his views on the salutary alternative they offer to the "verbal reasoning which has characterized the classical approach."
2. A reasonable source for a "standard" analog computer notation is that given in Blum (1968).
3. Our laboratory currently has one 15 amplifier analog computer of type EAI 380 with repetitive operation feature and X-Y plotter. In addition, we have a PDP-8L digital computer with teletype and A-D, D-A converter.
4. I have written a 30 page booklet on this aspect of programming in FOCAL (Howland 1971) as well as a FOCAL manual (Howland 1972a). At the time they were written, the FOCAL language appeared to offer the most computing power on a small computer, and I believe that FOCAL is still superior to most mini computer BASICs in power afforded the user.
5. Not all of our students' homework problems are done on the laboratory digital computer. Our students have access to the Division of Biological Sciences' interactive computing facility which offers FOCAL and BASIC at alternate times at four terminals on a reservations basis. I have described this facility elsewhere (Howland 1972).
6. We have found that Conte's text on numerical analysis (Conte, 1965) is very useful as a source for algorithms in this area.
7. It should be candidly admitted that some of these students are more knowledgeable in applied mathematics than their instructor. However, this is a common situation in physiology which, as one German physiologist observed, "has been getting too difficult for physiologists for the last 150 years." There are a large number of problems endemic to the teaching of physiology which stem from its position as the most derivative of all sciences. For example, teachers of physiology are faced with another enormous expansion of knowledge on the biochemical aspects of their profession.

BIBLIOGRAPHY

- Blum, Joseph J. (1968). Introduction to Analog Computation. Harcourt, Brace & World, Inc. New York.

- Coaroe, J. H. (1965). Physiology of Respiration. Yearbook Medical Publishers. Chicago.
- Coate, S. D. (1965). Elementary Numerical Analysis. McGraw-Hill Book Company. New York.
- Grodins, F. S. et al. (1954). Respiratory Responses to CO₂ Inhalation, A Theoretical Study of a Nonlinear Biological Regulator. J. Appl. Physiol. 7:283.
- Howland, H. C. (1972). A FOCAL Primer, 2nd ed. Algebraic Languages. Ithaca, New York.
- Howland, H. C. (1971). Solving Quantitative Homework Problems with FOCAL. Division of Biological Sciences Interactive Computing Facility. Cornell University. Ithaca, New York.
- Howland, H. C. (1972b). (In press). Algebraic Language Instruction in the Biological Sciences Curriculum at Cornell. Jr. College Science Teaching.
- Milhorn, H. T. Jr. (1966). The Application of Control Theory to Physiological Systems. W. B. Saunders Co. Philadelphia.
- Stewart, Peter A. (1970). Computers in Undergraduate Physiology Teaching. In Proceedings of Conference on Computers in the Undergraduate Curricula, Center for Conferences and Institutes, The University of Iowa, Iowa City.

COMPUTERIZED ECOLOGY SIMULATION

Ernest M. Salter
Cottey Junior College for Women
Nevada, Missouri

Gerald M. Pitts and Barry L. Bateman
University of Southwestern Louisiana
Lafayette, Louisiana 70501

Professors in the field of ecology have a difficult time in presenting to the undergraduate student the ideas that make up the concept of genetic-environment interaction which are basic to the study of ecology. At the University of Southwestern Louisiana, Lafayette, Louisiana, and Cottey Junior College for Women, Nevada, Missouri, a simulation model has been developed which allows a student to study such an interaction on a closed population of a single species or to predict the results of such an interaction for various contemplated environmental changes. The model can be activated by the professor or the student through one of the eight remote terminals located strategically around the campus (University of Southwestern Louisiana only, Cottey Junior College depends on distributed hard copy). By changing the input of individual factors (individual gene content and dominant and recessive gene factors) and/or the environmental factors (population number, number of predators, food supply, water supply, age death chart, and accident ratio) the student can receive direct feedback as to the genetic and environmental results after a selected number of years.

The problem of determining the time (in generations) for a genetic variation to become predominate is dependent on the weight factor which a certain genotype, or group of genotypes, contributes to the survival of the individuals. Since different rates of producing variation occur in nature, seemingly independent of the present, the model permits a variety of environmental factors to interact with the genetic pool of the population on a weighted factor basis. Thus the birth and death of individuals is based on the environment and the genetic make up of the individual.

A genetic examination of the simulation should provide some insight into the working of the model. The input consists of any number of individuals along with their gene patterns, tables for the mating and birth routines and tables for each of the death routines. Death is provided for by predators, lack of food, lack of water, age and accident.

In the mating and birth routines, each male in the population is given a number of chances to mate with a female, depending on the population size. For each successful mating, the litter size is computed from the genetic characteristics of the parents and a litter size table. The genetic make up of the newly born individual is determined by Mendelian choice from the parent genes. After a successful mating, the female is tagged to prevent mating again within the same year.

After mating is completed, the number of predators for two years in the future is computed. The number of predators is proportional to the number of individuals that they feed on, but the model uses a two year delay for the computed number of predators to be engaged in the model.

In the predator routine, the individuals die according to the genetic dominant or recessive character of genes which would have an effect on the predator avoidance of the individual. The genetic structure, along with the number of predators, determines the number of individuals dying by this routine.

The next death routines, lack of food and lack of water will be discussed together because of their similarity. The current food and water supply are changed according to the present population and a recovery factor based on the previous year's population. This is done in order to reflect the fact of nature that the more individuals there are the less food and water there exists per individual and the slowness of recovery of the food and water supplies. For the genetic characteristics which effect the ability of an individual to find or utilize food or water supply, a death number is calculated and used to determine whether the individual survives.

In the age routine, the death rate is determined according to the age of the individual and certain genetic qualities. The age table entered previously determines the death rate.

The accident routine provides for removal of individuals by a random, non-genetic, non-environmental basis by a percentage of the individuals in the system at that moment.

The output of the simulation consists of a summary of the number of individuals born and killed by each death factor each year. A sample can be seen in Table I. Every period of a selected number of years a summary of the total genetic characteristics of the population is given and a sample can be seen in Table II. From these tables, the student is able to follow a certain genetic characteristic change and the population as this genetic-environment interaction has proceeded.

TABLE I

INTERNAL STATISTICS

PERIOD	YEAR	NUMBER START	NUMBER BORN	NUMBER DIED	BY PREDATOR	FOOD	WATER	AGE	ACCIDENT	NUMBER PREDATORS	FOOD SUPPLY	WATER SUPPLY
2	1	42	28	19	6	1	5	7	0	6	1.52	4.52
2	2	51	61	26	8	6	5	7	0	7	1.53	4.53
2	3	86	13	16	4	2	5	3	2	10	1.52	4.53
2	4	83	97	46	12	12	7	13	2	16	1.52	4.55

AGE TABLE

PERIOD	YEAR	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
18	9	11	4	4	0	3	1	1	0	0	0	0	0	0
43	15	7	10	3	4	0	4	0	0	0	0	0	0	0
10	37	12	6	10	3	2	0	3	0	0	0	0	0	0
68	7	30	11	3	9	2	3	0	1	0	0	0	0	0

TABLE II

DOMINANT GENOTYPES PARTS PER TEN THOUSAND

MALE	GENE2	GENE3	GENE4	GENE5	GENE6	GENE7	GENE8
4402	6044	7014	8432	6865	7611	9029	7835
GENE9	GENE10	GENE11	GENE12	GENE13	GENE14	GENE15	
7761	6417	8283	5373	7611	7089	8582	

RECESSIVE GENOTYPE, PARTS PER TEN THOUSAND

FEMALE	GENE2	GENE3	GENE4	GENE5	GENE6	GENE7	GENE8
5597	3955	2985	1567	3134	2388	970	2164
GENE9	GENE10	GENE11	GENE12	GENE13	GENE14	GENE15	
2238	3582	1716	4626	2388	2910	1417	

TABLE II (CONTINUED)

DOMINANT GENOTYPES, PARTS PER TEN THOUSAND

GENE2	GENE3	GENE4	GENE5	GENE6	GENE7	GENE8	GENE9
3619	4291	6268	3992	5597	6716	5074	4962
GENE10	GENE11	GENE12	GENE13	GENE14	GENE15		
3768	5671	3395	5149	4776	6007		

RECESSIVE GENOTYPE, PARTS PER TEN THOUSAND

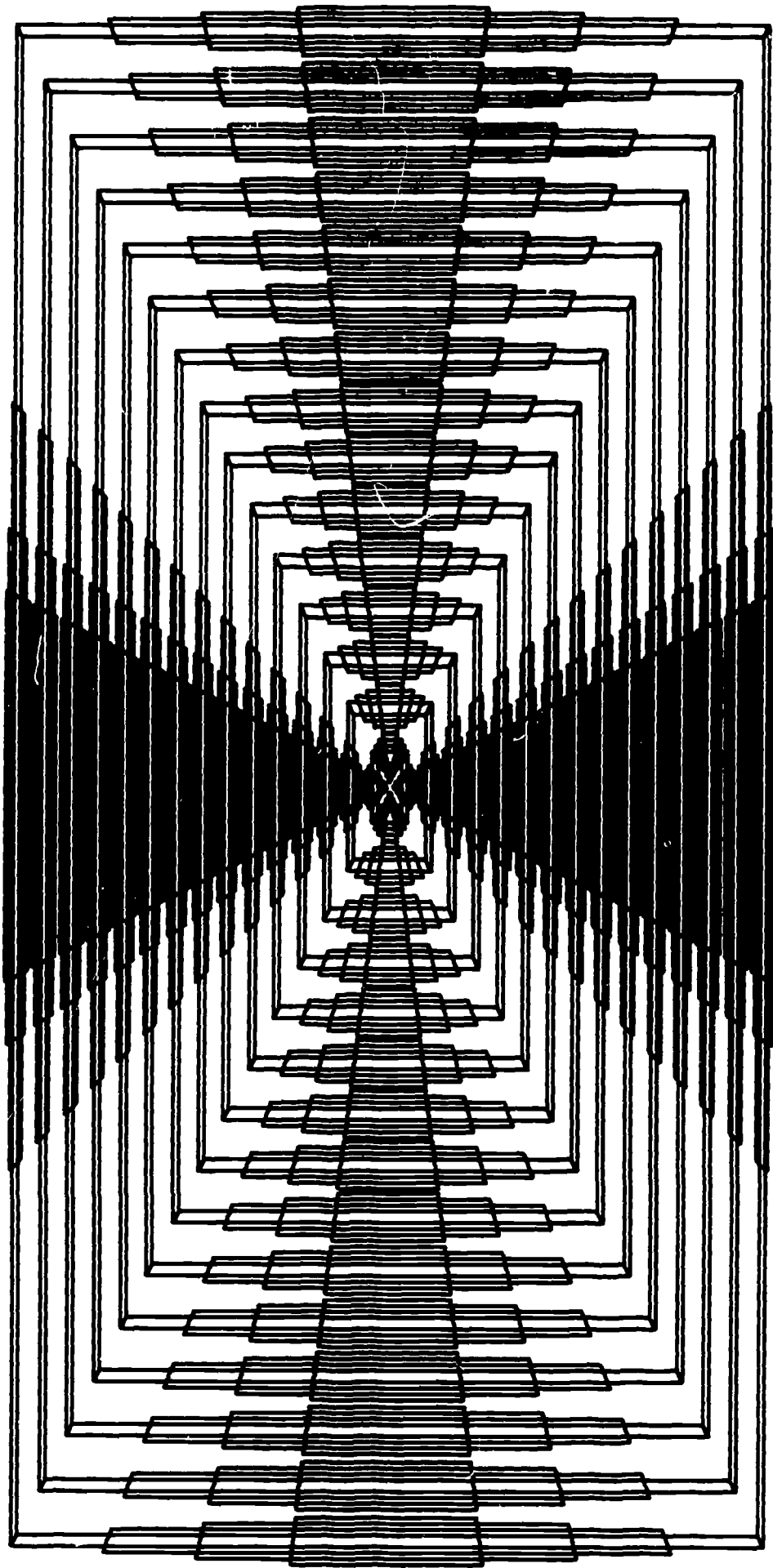
GENE2	GENE3	GENE4	GENE5	GENE6	GENE7	GENE8	GENE9
6380	5708	3731	6007	4402	3283	4925	5037
GENE10	GENE11	GENE12	GENE13	GENE14	GENE15		
6231	4328	6604	4850	5223	3992		

The model has permitted arbitrary selection of environment and genetic weight factors. This has allowed the users to examine and re-evaluate the results with the major features of interaction presently documented. It also has permitted the users to make predictions of variations likely to be produced if the environmental and genetic factors of the species can be reasonably determined.

Primarily, the validation of the program was handled by the Biology Departments of the University of Southwestern Louisiana and Cottey Junior College for Women[1,2]. This model gives the student a "feel" for the genetic development over a number of years under varying environmental conditions that could not be obtained through classroom lecturing or laboratory work. The student has the capability of answering his own questions by simply posing the questions to the computer in the form of specific input parameters and receiving immediate feedback as to the effects.

REFERENCES

1. Cordes, Private Communication, University of Southwestern Louisiana, Lafayette, Louisiana, November 17, 1971.
2. Goering, D. K., Private Communication, Cottey Junior College for Women, Nevada, Missouri, November 29, 1971.



Jim Durford

PERSPECTIVE #2 by Jim Durford
Problem: Diminishing Polygon Forms

COMPUTER BASED INQUIRY INVESTIGATIONS IN BIOLOGY

James C. Horton, David S. Hinds and Mary Ellen Burrows
California State College
Bakersfield, California 93309
Telephone: (805) 833-2123

All courses in Biology at California State College, Bakersfield, are taught in the inquiry method. Students are required to take an elementary course in computer programming, and this ability is later used for creating models, and for analyzing and summarizing data. We employ the computer in another fashion to overcome the resistance of students to employ arithmetic to prove formulas or to become familiar with the workings of a mathematical relationship. We have devised several programs which may be used by students without any previous knowledge of computer equipment for the specific purpose of extending a formula into areas not given in the text. The student is able to apply a concept, quantified by the formula, to various situations of his choosing and thereby validate the expression in each specific case. When groups of students use this technique for a number of different examples, a later class discussion can be used to correlate the extension of a formula over a wide range of situations, verifying its application without a large expenditure of individual effort. In this way, mathematical expressions become a part of the student's vocabulary because he has used them extensively rather than accepted them by rote.

As an example, the following formula for rabbit-fox predation came anonymously to the author. It has been used successively in several courses involving predator-prey relationships, in the use of mathematical modeling to predict outcomes of population interactions, and as an example of population dynamics.

$$x_1 = x_0 + (Ax_0 - Bx_0y_0)t$$

$$y_1 = y_0 + (Cy_0x_0 - Dy_0)t$$

Verbally the formula states that the number of rabbits present at a given time (x_1) is equal to the number of rabbits found initially (x_0) plus rabbit natality (Ax_0) less those rabbits preyed upon by foxes (Bx_0y_0). Similarly the foxes present at the same time (y_1) are equal to the initial number of foxes (y_0) plus fox increases resulting from reproducing parents existing on rabbits (Cy_0x_0) minus those dying of starvation (Dy_0). The formula is simple enough to be readily understood by most students and is still capable of considerable management. The original values in the formula of A, B, C, D are 4, 2, 3, 1 respectively, x_0 was 6000, y_0 was 2500 and t was 0.01 with a printout every 12 cycles. The expression obviously is curvilinear so an approximation of the curve was obtained by re-establishing new values of x_1 and y_1 at unit intervals of time (0.01) and printing each twelfth approximation. (For convenience in our program every tenth value was printed.) It is possible of course to consider each unit time interval as a generation, but we found it more convenient to regard each tenth interval as a generation time. (A copy of a publication given to the senior author included the above information as well as a program for calculation, for printout and for a display of the output as a graph. Unfortunately, the copy contained neither author nor publication source. Hence we are unable to express proper credit, but it probably should be attributed to L. B. Slobodkin.)

Printouts of tenth intervals were plotted against population size by the student. Two data cards were supplied, one for the constants A, B, C, D and one for initial population size. Once the program was operational, duplicate decks of cards were provided to groups of 2 or 3 students. Local stored programs would be better but in the absence of easy access to this method, card decks were numbered to prevent mis-operation due to shuffling. Individual student groups were identified by separate job cards and they punched their own data cards. Persons without experience at the keypunch machines asked their colleagues to illustrate how these might be used, and all students had an opportunity to punch their own cards.

Our first use of this particular problem in an environmental population biology course required that students apply the formula expression using different population sizes. Fixed data cards for the constants were used and students were allowed to vary population sizes between one and ten thousand. The figures they selected were individual and each group plotted their printouts. Comparisons were made in a discussion session. After an early and initial variation, the students were surprised to find that curves for all populations were remarkably parallel. From this they reasoned (with help) that the relationship dictated by internal constants forced a pattern of fluctuation which was not overcome by population size. This indeed, was the first point we wished to make. It is our opinion that the students were more aware of this condition by being forced to discover it for themselves, than if we had told them this was so.

The second step was to ask the students to predict those levels of fox-rabbit populations which would be stable over time. The students commonly selected an intersection of fox-rabbit curves from their graphs and then placed these values in their program. Again the curves paralleled those generated in the first output and the students were once more forced to recognize that population interactions dictated that these situations would not become static but remain in some kind of a variable balance.

The third step was to allow students to change the internal constants at will. In this case various proposals suggested the direction of the change e.g., suppose something increased or decreased the rabbit birth rate (A), somehow success of fox predation varied (B), the fox natality or birth mortality changed (C) or fox nutrient requirements were altered (D). In groups, students were asked to generate reasons for changes in a constant and then to create a series of curves by plotting the resultant populations against the original (Table 1). Groups increased or decreased a particular constant so that all variations might be explored. These comparisons provided insight of how minor changes could result in drastically altered population patterns. Students speculated on situations which might bring about changes in constants and in a fourth step, to predict the consequences of this change. A further condition was established in which students were asked to predict the conditions required for stasis. The output allowed verification of each student prediction and although none resulted in the static condition, the students were aware of the multiplicity of factors influencing any population level.

From these kinds of inputs, further discussion led the students into exchanges concerning dynamic balances, population interaction, the cyclical nature of population variability, and the difficulty of estimating correct constants. Without the stimulus of a program which they could manipulate, the students would invariably have reiterated the formula and would have known little of its operation. The problem selected was almost juvenile in its simplicity and yet the individual flexibility to predict, to manipulate the variables and constants, and to plot the outcome produced in the students an unusually wide recognition of the forces implicit in population interactions.

A similar project involved the Hardy-Weinberg formula in a genetics course. This formula which is an expansion of a simple binomial predicts gene frequency in a randomly mating population where gene changes do not take place, where the population remains constant, and immigration or emigration does not occur. The Hardy-Weinberg equation is:

$$(p + q)^2 = 1$$

where p, q are frequencies of two alleles of one gene and where $p + q = 1$. The formula is widely used in all genetics courses and without exception, the students memorize the formula, and apply it in situations without, perhaps, fully understanding its import. Using the same idea as mentioned above, a program was concocted to provide the arithmetic machinery and printout with the data cards being the only variable to be punched by students. At first students were asked to verify that the formula does indeed predict gene frequency over long periods of time (50-100 generations) using individually selected gene frequencies and population sizes. Perhaps this exercise was redundant in that students invariably received printouts with the same gene frequency for the lengthy period and it is suggested that a shorter period of time might be more profitable.

Once the students were convinced that these predictions were true, variations were introduced. Gene frequency for each generation was changed to simulate the process of gene mutation and the resulting changes in gene frequency were plotted. Selections for or against certain gene combinations were programmed and shifts of gene frequency were again plotted. The influence of migrants entering the population with different gene frequencies was included and lastly, gene drift was introduced using a random number generator and was used with different gene frequencies. In each case, the students selected constants which were in the usual range encountered and were required to plot the effects of these gene changes over a period of 50-100 generations.

In a large sense, much of a genetics or population dynamics course can be built around the programming. From those cases where population gene pools were constant, (equal to conditions encountered on a small island) students were asked to illustrate the "founder principle" and to relate its effects upon ensuing generations. Natural selection and evolution of traits were included in this particular discussion and the students became aware that a gene pool consists of a large number of individuals and breeding randomly. The next variation introduced was mutation and students were able to plot changes in population size (in terms of gene frequency) as a consequence of forward and back mutation rates. Because of the ability to see the effects of mutations over a large number of generations immediately, the students were aware of the slow rate of gene frequency change under normal rate conditions (10^{-4} to 10^{-8} /generation). Students would not normally be exposed to this visible evidence since they are reluctant at least, to

Table 1. An example of printout resulting from changes in the variables B and C in the fox-rabbit predation formula. Only the maxima, minima and amplitude from each situation are given. The figures represent the change in each population from original values of 6000 rabbits (x_0) and 2500 foxes (y_0). The two curves are not superimposed and the time lag is that of the fox curve behind (in response to) the rabbit curve.

A	B	C	D	Maximum Rabbit	Minimum Rabbit	Rabbit Amplitude	Maximum Fox	Minimum Fox	Fox Amplitude	Time Lag	
4	2	1	3	6.186	1.168	5.019	3.759	0.892	2.867	0.320	(original)
"	3	"	"	7.720	0.773	6.947	3.042	0.421	2.621	0.290	
"	4	"	"	9.827	0.445	9.382	2.819	0.200	2.619	0.260	
"	5	"	"	12.164	0.243	11.291	2.716	0.098	2.618	0.240	
"	6	"	"	14.634	0.129	14.505	2.658	0.049	2.610	0.220	
4	2	1	3	6.186	1.168	5.019	3.759	0.892	2.867	0.320	(original)
"	"	2	"	6.046	0.124	5.992	6.756	0.248	6.508	0.240	
"	"	3	"	6.020	0.018	6.003	9.557	0.072	9.485	0.190	
"	"	4	"	6.010	0.003	6.008	12.171	0.021	12.150	0.160	
"	"	5	"	6.005	0.000	6.005	14.645	0.006	14.639	0.140	

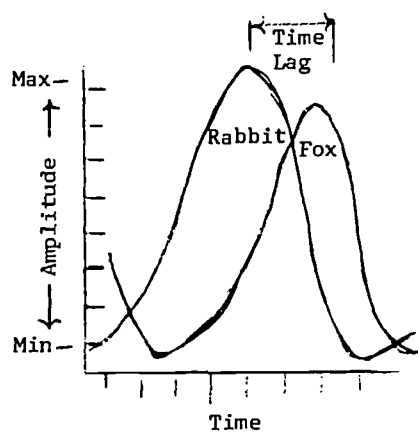


TABLE 1.

carry out 100 separate arithmetic binomial expansions involving a slightly different gene frequency change in each expansion. In the next example where selection against or for a specific genotype occurred, students were able to see immediately the effects of selection. Populations, in terms of gene frequency, decreased or increased markedly illustrating the principles so often iterated to undergraduate students and so least often understood. The additional complicating factor of immigrants and emigrants in the population was also illustrated graphically, particularly when they were allowed to vary the number of migrants per population cycle.

Once these ideas were well in hand, the value of the computer was reinforced when students were able to put together two or three of these separate changes to gene frequency in the population. Mutation under conditions of specific selection for the mutant, migrants with selected-for or selected-against characteristics, closed populations with selection against the recessive, and the emergence of dominant forms served to illustrate more dramatically to students the changes in population than lectures ever could have done. The full implications of this type of programming in the teaching of genetics courses has not yet been explored, but the possibilities seem endless. Costs per student in terms of computer time were approximately \$0.16 per program and each student (working in groups) ran 2-3 programs.

Numerous other examples in quantitative biology are susceptible to immediate application of this mode of computer inquiry. We have initiated examples of lung-gill-oxygen exchange, muscle-bone-leverage principles, diffusion over cell gradients, and the integration of taxonomic similarities in programs which allow students to explore the variabilities allowed in a mathematical model of a system, and to relate these studies to physiologic, anatomic, and thought problems. We feel confident that these programs which provide students with an opportunity to explore the parameters dictating a dynamic balance or organismal capability will play an important part in enticing student involvement and elucidating hitherto obscure relationships. We feel that an important portion of the educative process is in the incorporation of a working relationship of predictive formula into the student's personal body of knowledge. In most cases these relationships can be displayed most economically through the use of computer programs. In the comparison of 100 generations of fruit flies versus 100 generations of computer printout there is little doubt of economy. Similarly that these relationships should become a working part of the student's vocabulary there is little doubt. We suggest this method as worthy of experimentation by other institutions and intend to employ it further in our inquiry investigations.

COMPUTER ASSISTED ALGORITHM LEARNING IN ACCOUNTING

William F. Bentz
The University of Kansas
Lawrence, Kansas 66044
Telephone: (913) 864-4665

Introduction

The purpose of this paper is to present several hypotheses about the potential advantage of learning accounting methods with the aid of computers. A synthesis of relevant learning theory principles forms the conceptual foundation for these hypotheses, and several examples serve to illustrate computer assisted instruction. The focus is only on those accounting methods which can be characterized by computer algorithms. Due to space limitations, many other equally important aspects of accounting instruction cannot be considered here.

Background

While many innovative applications of computer technology have been developed for accounting instruction purposes, almost no attention has been given to learning theory concepts implicit in these applications. The development of computer assisted instructional (CAI) [1] materials for accounting is apt to be both haphazard and inefficient until a conceptual model of the relevant learning processes has been developed. More precisely, we must have a conceptual model of learning processes in mind in order to

- (1) set specific instructional objectives that serve to guide the educational process[2];
- (2) formulate hypotheses, based on learning theories developed in other contexts, about the expected contributions of alternative instructional methods to the set of instructional objectives;
- (4) efficiently select CAI materials, based on their hypothesized benefits as indicated by theories of learning, for further development and experimental testing; and even to
- (5) investigate successful applications of CAI to identify the important elements of these applications and to determine the nature of the success achieved.

In addition to having a weak conceptual foundation, many educators adopt such a limited view of CAI that its development may be unnecessarily constrained. In accounting, the dominant view of the computer is that it is a giant calculator. Thus, computers are viewed as a particular type of instrument which serves one function - calculating [3]. A more inclusive view is that computers are valuable teaching aids that can facilitate the learning process in many ways, in addition to performing purely calculating functions[4]. The instrument view of computers tends to limit their use to beginning courses, while a broader perspective is more likely to result in the development of CAI materials at all levels of instruction.

The Structure of Accounting

The superstructure of accounting has been studied by several researchers, including Ijiri [16], Mattessich [18] and Sterling [23]. These efforts involve attempts to describe accounting, as practiced today, in as compact a manner as is possible. There are at least two practical benefits of such efforts. First, by capturing the essential structure of accounting in a few axioms or laws, one can better communicate the essential characteristics of accounting to those outside the discipline; and, secondly, one can more efficiently describe accounting to prospective accountants.

However, at another level of abstraction, the structure of accounting can be viewed in a less formal, and more limited way. The subject of accounting is frequently explained by first partitioning it into topical areas which are deemed to have a structure of their own. For example, depreciation accounting, accounting for leases, accounting for pension plans and many other topics are discussed somewhat independently, in spite of the common superstructure which describes all of financial accounting.

By shifting our focus from the superstructure of accounting to a lower level of abstraction, accounting can be viewed as a collection of algorithms which relate to particular topic areas. "An algorithm is a procedure for solving a problem." [15, p. 1] In more formal terms, an algorithm is something that can be carried out on an idealized machine, called a Turing machine. Although the properties of Turing machines have been more precisely defined, let it suffice to note that a Turing machine can do anything a stored-program computer can do [15, p. 170]. Therefore, for our purposes those procedures that can be performed by a stored-program computer are called algorithms.

The Importance of Algorithms in Accounting

Many accounting techniques can be characterized as algorithms; thus, the learning of accounting techniques involves the learning of algorithms. Typically, the student reads through a demonstration problem and notes the sequence of steps used therein. Then, the student works several problems, mimicking the sequence of operations that are executed in the demonstration problem. The sequence of operations (an algorithm) necessary to solve an exemplar of a class of problems becomes apparent to the student during the process of examining the demonstration problem, or while working the homework problems. If the method is understood, it is understood as a technique which is applicable to problems other than the particular one at hand. Thus, the student who understands the method can apply it to a new problem with little difficulty, while the student who rote learns a sequence of operations to solve problem X may have difficulty solving problem Y, even though X and Y are of the same class of problems. Although other types of learning may take place while one works problems, the learning of accounting algorithms, and the practice involved in applying them to specific situations, are the major functions served by procedural-type homework problems.

There are two additional reasons for focusing on accounting algorithms. In auditing, the professional accountant is faced with the problem of evaluating the system of internal controls that assures the quality of the accounting data on which an opinion must be rendered. Because many accounting records and recording processes are being automated, modeling or flowcharting the processing system is of utmost importance on almost every audit. Part of what is being modeled or flowcharted is the algorithm that represents the method by which a machine transforms inputs into output. Therefore, the auditor must be prepared to deal with algorithms in their general form, not with a particular solution method for test problem X. Precisely because the clerical procedures performed inside the machine cannot be observed, the modeling of these unobservable algorithms is more important than ever before.

The construction and testing of algorithms is an equally important function of the auditor's counterpart, the management consultant. In designing a new system, or in revising an existing system, the management consultant must specify how activities are to be accomplished. At some stage in the design process, a designer must specify the detailed procedures to be executed by the system. This specification is, in essence, an algorithm even though the system being designed may not be an automated system. Thus, for management consultants, the specification of accounting algorithms is an integral part of the system design process.

Learning Accounting Algorithms

Now that the importance of looking at the algorithmic nature of accounting methods has been discussed, we can focus on three approaches to learning accounting methods: (1) The traditional problem-solving approach, (2) reception learning of generalized algorithms, and (3) discovery learning of generalized algorithms.

The traditional method. In freshman through junior level courses, the instruction sequence usually includes the presentation of some concepts, propositions, and background information. This material is followed by the working of accounting problems which serve to clarify and illustrate the concepts presented and the accounting methods involved. As mentioned in the previous section problems involving a sequence of operations tend to be solved in one of two general ways. First, if a demonstration problem is presented, the student tends to examine the demonstration problem and then attempts to work a problem, or he attempts to work a homework problem using a demonstration problem as a guide. In either case, the learner is actively attempting to discover for himself the sequence of arithmetic operations involved in the accounting method. Understanding of the method can be regarded as complete when the student can apply the method to similar problems (application), or even to novel situations which have not been related to the technique before (problem-solving learning as described by Ausubel[4]).

Several criticisms of the "problem" method of learning implicit algorithms are in order. First, since the student never sees the algorithm in its general form, he may stumble through several problems before understanding the methodology embodied in the undisclosed algorithm. Second, the slower learner may not receive sufficient feedback from a few homework assignments

to fully grasp the method. Third, the learning of a method by working problems may only result in pre-verbal understanding which cannot be retained as long as a verbally created description. Fourth, even if a student learns a method by working problems and can verbalize that understanding, he may have difficulty remembering the more important features of the method unless he has the opportunity to work with an abstract model of the method. Further, the more general and clearly defined algorithms should be more easily remembered than a collection of less general methods which are not clearly differentiable[4, Chapter 5]. Fifth, there is a tendency to teach business terminology and business practices by introducing them in assignment problems. While learning about business practice is important, the superimposing of concept learning, proposition learning and problem-solving may only confuse the student and impede the learning of accounting algorithms.

Reception learning of generalized algorithms. A second way of learning accounting methods is to encounter them in complete form, and then to apply the method, or algorithm, to particular problems. The sequence of instruction would include a presentation of concepts, propositions and background information, as before, followed by a presentation of the accounting algorithm in general form. After the algorithm is studied, it is applied to a series of problems which serve to clarify the students' understanding of the algorithm, as well as demonstrating the range of applications associated with the algorithm being studied. Each student's understanding of the algorithm is tested in the same manner as described above.

Algorithms can be presented for reception learning by means of several different devices. Flowcharts, decision tables and computer programs coded in procedure-oriented languages can be used to describe almost any accounting procedure. As computer courses become required, these tools become more and more familiar to all business students, so their classroom use is feasible in many schools.

Note that these two methods of learning are very different. In the first case, the algorithm must be inferred from an example, or from the feedback provided while the student is attempting to solve problems "correctly." In the second method, the student learns the algorithm by reception, rather than by discovery. The algorithm is presented in its final form, so the student need not discover it for himself.

There are several criticisms of the reception learning method, as there were of the traditional method. First, students may apply the algorithm to specific problems in a rather mechanical manner, which may not require them to think about the method itself. Second, even when students think about the algorithm as a generalized technique, they may attain only a preverbal understanding of the algorithm. Preverbal understanding is only an intermediate phase of the learning process and does not represent a terminal learning objective. Third, the act of applying a specified algorithm to a set of problems may lack the motivational qualities of a problem or puzzle, that must be solved.

Presenting accounting methods as algorithms does have several advantages which alleviate the limitations of the traditional method mentioned above. First, by focusing on the algorithm itself, the instructor is telling the student what is important. In the process of solving individual problems without knowing that a general solution method exists, a student may not be able to separate the important features of a homework problem from the trivial. Secondly, by working with a defined algorithm, rather than attempting to discover the algorithm, a student may be able to learn more about its structure and essential features because it has been fully specified for him, in general form.

Another important reason for present algorithms is probably clear to most readers. Some ideas are simply vague and somewhat ill-defined until expressed in equation form, or as algorithms. For example, the time value of money is a concept which is readily accepted by students on an intuitive level when first explained. After a careful and tedious presentation of present value concepts, many students can handle straight-forward interest problems with the use of interest tables, but only the students with some mathematical sophistication seem to grasp the process of discounting a series of payments to determine their present value. The ability to work with symbols and equations seems to be necessary if a student is to understand interest problems.

In summary, presenting accounting methods to students in the form of algorithms has certain advantages over the more traditional problem solving approach. Specifically, it is hypothesized that understanding is more nearly complete and retention is greater when students study and apply explicitly algorithms, rather than discovering accounting methods by solving problems. This hypothesis is based on Ausubel's theory of the learning and retention of meaningful material.

However, there are some disadvantages associated with reception learning. The ways in which CAI can be expected to alleviate these disadvantages are discussed next.

Computer assisted discovery learning of accounting algorithms. An alternative to reception learning is the discovery learning of accounting algorithms. In contrast to solving a sequence of problems, the assignment is to construct a general algorithm which then can be used to solve a whole class of accounting problems. Text material, illustrative problems and handouts can be used to specify the accounting method, but the student is forced to generalize the method so that an algorithm can be formulated. The algorithm can be characterized by flow charts, decision tables, or an operational program coded in a language such as BASIC, FORTRAN, or COBOL.

What are the potential benefits of having students reconstruct accounting algorithms by writing computer programs? First, consider the potential motivational benefits. In the process of writing a program, the student receives a lot of feedback which tends to support continued work on the problem (a hypothesis). The feedback is in the form of error listings from the computer and the computed answers to a test problem. If a student gets an incorrect answer, he has a clear signal that his algorithm contains some errors. Discussion with the other students that congregate at the computer center and comments by the instructor also serve as valuable feedback. Incidentally, it is relatively easy to follow the logic of a student's program when the procedure being programmed is a familiar one and when a list of suggested variable names has been provided as part of the assignment.

Second, people tend to think about incomplete tasks more than they do about completed tasks[4, p. 498]. Therefore, it is hypothesized that working on one computer program for N days may maintain more concentrated student attention than working and completing several different problems over the same time period. Another motivational aspect is the greater opportunity for satisfying ego-fulfilling needs when the student must construct an algorithm which is not presented to him in completed form. Constructing the algorithm is satisfying in itself, and learning computer skills is satisfying to many students because of the career opportunities associated with a knowledge of computers.

One final motivational benefit is the opportunity to complete satisfactorily a task before submitting it for final approval. Some students find it very frustrating to spend hours working complex accounting problems, to achieve only partial success. With programs, if enough lead time is provided, diligent students have the opportunity to write satisfactory programs.

The potential cognitive learning benefits of the process of constructing accounting algorithms are dependent on the claim that the student will thoroughly understand an accounting procedure if he has written a computer program for it. Further, it is hypothesized that understanding an algorithm is a higher level of abstraction than can be achieved by most students in the process of solving particular problems. To the extent that these claims are true, a student's knowledge of the accounting procedure that he programmed should be integrated into his cognitive structure as a generalized method which is clearly differentiable from other accounting methods.

There are several conditions which can be expected to facilitate greater cognitive learning behavior. First, the construction of an algorithm requires more active participation in the learning process than does the learning of an algorithm presented in the final form (reception learning). Some students will critically evaluate new ideas to "make sense" out of them while working at the task of incorporating them into cognitive structure. However, all too many students are passive listeners when propositions or problem solving methods are presented to them in final form. Therefore, to the extent that more active learning can be induced by requiring students to construct algorithms, it is hypothesized that greater learning is to be expected. Experience indicates that most students do need some inducement to become actively involved in the learning process.

Second, the benefits of massed learning, as opposed to distributed learning, are related to the learning of algorithms[5]. In this case, the alternative learning methods being compared are the learning of accounting methods by solving individual problems as opposed to constructing algorithms for subsequent application. In solving a sequence of problems, usually of increasing difficulty and complexity, the student encounters two problems: forgetting between problem-solving sessions, and the warm-up required to recall the methods and settle into the new problem. It seems plausible that forgetting can be a factor since most undergraduate students take five or six different courses each semester, plus working and being involved in other activities. Further, since accounting problems usually represent complex tasks, substantial warm-up and reorienting of one's thinking may be required each time a new accounting problem is encountered. Under these conditions, massed learning can be more efficient than learning distributed over a number of sessions.

Therefore, to the extent that there is a sizeable threshold of effort required to discover and fully grasp accounting algorithms, the massed learning that is usually associated with the writing of a computer program may be more efficient than the solving of a series of individual problems over time. Because of the difficulties created by forgetting between problem-solving sessions, and the warm-up required to settle into a new problem, it is being hypothesized that

the benefits of massed learning are applicable to the construction of computer programs, thus improving learning.

Greater lateral transferability of knowledge should also be facilitated by having students write programs. Gagne[12, p. 235] theorizes that applying one's knowledge in a number of different contexts increases transferability, although we know little about the precise factors which are involved. A student can use his own program, or a previously written program, to solve a variety of problems, thus emphasizing the generality of the techniques without increasing the busywork that is usually associated with working a large number of problems. Thus, a student can be encouraged to generalize his conception of a technique.

Illustrations of Computer Assisted Instruction

Process cost accounting, the allocation of service department costs among reciprocally dependent service departments and operating departments, the allocation of profits among reciprocally owned corporations, corporate budgeting models, and the financial accounting methods which involve present value calculations are all complex topics which involve algorithms. For process costing, students can be given a set of program specifications in order to write programs which accept standardized inputs, and generate the required cost reports. Students can also be asked to specify how the input data is to be collected, and how cost reports are to be designed and distributed within a fictional company.

Service department cost allocation techniques, profit distribution techniques, and corporate budgeting models all involve using matrix operations to solve systems of simultaneous linear equations. Their common structure becomes quite apparent to students when they program these techniques. Similarly, the common structure of lease contracts, pension plans, bonds, and long-term investments becomes apparent to students who have constructed an algorithm to find the present value of a single payment or of a series of payments. Moreover, students seem to understand the present value techniques much better after having written such a program.

SUMMARY

In this paper, several learning theory concepts have been related to algorithm learning in accounting. An examination of the relevant learning theory concepts leads to the hypothesis that appropriate CAI techniques will facilitate the learning of accounting algorithms.

FOOTNOTES

1. No distinction is made between computer assisted instruction (CAI) and computer extended instruction (CEI) here.
2. The importance of setting measurable objectives in education is well accepted as is demonstrated by the comments of Churchman[6 and 7], Goldiamond[13], Homme[14], Evans[10], Gagne[12], and Ausubel[4].
3. The instrument view of computing is found in Anderson[3], Beams[5], Mastro[17], Mecimore[19], Penick[20]. Person[21], Corcoran[8], and even the committee reports of the AAA[1] and the AICPA[2].
4. See Cowie and Fremgren[9], Frank[11], Prater[22], and parts of[1].
5. For a discussion of massed learning of low-level capabilities, see Stanley Stevens[24, pp.636-40], and Robert S. Woodworth and Harold Scholsberg[26, pp. 786-94].

REFERENCES

1. American Accounting Association Committee (1964) on Courses and Curricula--Electronic Data Processing. "Electronic Data Processing in Accounting Education," The Accounting Review, Vol. XL, No. 2 (April, 1965), pp. 412-28.
2. American Institute of Certified Public Accountants, Report of the Committee on Education and Experience Requirements for CPAs. New York: American Institute of Certified Public Accountants, Inc., 1969.
3. Anderson, John J. "Integrated Instruction in Computers and Accounting," The Accounting Review, Vol. XLII, No. 3 (July, 1967), pp. 583-88.

4. Ausubel, David P. and Floyd G. Robinson. School Learning: An Introduction to Educational Psychology. New York: Holt, Rinehart and Winston, Inc., 1969.
5. Beasm, Floyd A. "EDP and the Elementary Accounting Course," The Accounting Review, Vol. XLIV, No. 4 (October, 1969), pp. 832-36.
6. Churchman, C. West. The Systems Approach. New York: Dell Publishing Company, 1968.
7. Churchman, C. West. "On the Design of Educational Systems." Working Paper No. 86. Center for Research in Management Science, University of California, Berkeley, 1964. (Mimeographed)
8. Corcoran, A. Wayne. "Computers Versus Mathematics," The Accounting Review, Vol. XLIV, No. 2 (April, 1969), pp. 359-74.
9. Cowie, James B. and James M. Fremgren. "Computers Versus Mathematics: Round 2," The Accounting Review, Vol. XLV, No. 1 (January, 1970), pp. 27-37.
10. Evans, James. "Behavioral Objectives Are No Damn Good," in Aerospace Education Foundation, Technology and Innovation in Education. New York: Frederick A. Praeger, Publishers, 1968.
11. Frank, Werner. "A Computer Application in Process Cost Accounting," The Accounting Review, Vol. XL, No. 4 (October, 1965), pp. 854-62.
12. Gagne, Robert M. The Conditions of Learning. New York: Holt, Rinehart and Winston, Inc., 1965.
13. Goldiamond, Israel. "Motivation--Some Ways and Means," in Aerospace Education Foundation, Technology and Innovation in Education. New York: Frederick A. Praeger, Publishers, 1968.
14. Homme, Lloyd. "A Behavioral Technology Exists--Here and Now," in Aerospace Education Foundation, Technology and Innovation in Education. New York: Frederick A. Praeger, Publishers, 1968.
15. Hull, T. E. Introduction to Computing. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1966.
16. Ijiri, Yuji. "Axioms and Structures of Conventional Accounting Measurement," The Accounting Review, Vol. XL, No. 1 (January, 1965), pp. 36-53.
17. Mastro, Anthony J. "EDP in One Elementary Course," The Accounting Review, Vol. XLII, No. 2 (April, 1967), pp. 371-74.
18. Mattessich, Richard. Accounting and Analytical Methods. Homewood, Illinois: Richard D. Irwin, Inc., 1964.
19. Mecimore, Charles D. "Integrating EDP into the Elementary Accounting Course," The Accounting Review, Vol. XLIV, No. 4 (October, 1969), pp. 837-39.
20. Penick, Jack G. "ADP Equipment as an Accounting Teaching Tool," The Accounting Review, Vol. XLI, No. 3 (July, 1966) pp. 549-51.
21. Person, Samuel. "The Integrated Use of Data Processing Equipment in Teaching Accounting Subjects," The Accounting Review, Vol. XXXIX, No. 2 (April, 1964), pp. 473-75.
22. Prater, George I. "Time-Sharing Computers in Accounting Education," The Accounting Review, Vol. XLI, No. 4 (October, 1966), pp. 819-25.
23. Sterling, Robert R. "An Explication and Analysis of the Structure of Accounting." Working Papers No. 22 and 23 (Part Two). Lawrence, Kansas: The School of Business, The University of Kansas, 1969.
24. Stevens, Stanley S. Handbook of Experimental Psychology. New York: John Wiley & Sons, Inc., 1951.
25. Williamson, J. Peter. "The Time-Sharing Computer in the Business School Curriculum," Paper presented at the Kiewit Conference, Dartmouth College, Hanover, New Hampshire, June, 1971.
26. Woodworth, Robert S. and Harold Scholsherg. Experimental Psychology. New York: Holt, Rinehart and Winston, 1965.

THE BUSINESS CORE INTEGRATOR
AT INDIANA UNIVERSITY

Thomas L. Guthrie
Indiana University
Port Wayne, Indiana 46805
Telephone: (219) 483-8121

In the academic year 1968-69, faculty at the School of Business, Indiana University, Bloomington Campus, began experimentation with a thoroughly revised undergraduate curriculum which included what has been called the four course integrative core. The core is taken by first semester Junior class standing business students who meet specific prerequisites. The core consists of three principles courses in the functional areas of finance, marketing and production and a new course, Simulation of Business Enterprise. The conception and development of the core by the faculty, spearheaded by Dr. William B. Panacher, reached such a stage of maturity by the academic year 1970-71 that the American Association of Collegiate Schools of Business awarded Indiana University School of Business the prestigious Western Electric Fund's award for educational innovation in higher education for its outstanding undergraduate core program. The purpose of this paper is to describe the objectives of the integrative core, demonstrate the procedures involved, and relate the experiences and reactions to date, with primary emphasis being given to the unique part that the Simulation course plays in the core.

Objective

Given the overall objective of the Business school to graduate students capable of contributing to society in general and the business community in particular, the immediate objectives of the four-course integrative core are:

1. to provide the junior year business student with a rigorous and integrative education in the functional areas of business.
2. to provide a simulated business experience that will cause students to begin to think like businessmen, to identify themselves with business and to increase their enthusiasm for a business career.
3. to provide for the student a small group atmosphere facilitating the change from a passive educational experience to one of involvement and action.
4. to provide the student the further advantage of small size classes and a closer student faculty relationship in at least one of the four core courses (4).

Prerequisites to the core, in addition to Junior class standing, include the following:

	<u>Semester Hours</u>
Principles of Economics.	6
Economic Statistics.	3
Finite Mathematics	3
Calculus	3
Introductory Psychology.	3
Principles of Sociology.	3
Management Accounting.	6
Legal Environment of Business.	3

These prerequisites, which are strictly enforced, provide the student with a higher level of competency and consistency than had been required heretofore, thus allowing a more rigorous treatment of subject matter in the basic functional areas of business.

Implementation Overview

The Simulation course is paramount in meeting the objective to integrate the students' initial work in business decision-making. A computerized business simulation game is utilized in the course which is, of course, nothing new, since numerous business courses use computerized games of one type or another. What is unique is the relative importance of the game in the

course. In most business courses games are used as sidelights to demonstrate principles or to reinforce desired learning patterns. In the Indiana Simulation course the game is literally the course, and sidelight assignments are made in conjunction with the other three courses.

To administer the game, the students are divided into teams of 4 to 8 people, each, which comprise individual companies for gaming purposes. This is, again, not unique, but what is unique is that (1) individual team members attend the same section of all four core courses and (2) all four core courses must be taken concurrently.

By utilizing the team approach and the business simulation game as a central classroom activity, demanding the knowledge, analytical tools and methods taught in the other three courses, integration is achieved in several ways:

First, as a result of each team being in the same section of finance, marketing and production, each student is a member for the entire semester of a small management group which is responsible as a group for several assignments. The management group is bound together by common goals, especially with respect to the Simulation course. This, in effect, provides every student with 3 to 7 counselors, tutors and friends, business and social. The advantages of this type of atmosphere should be self-evident in these days when many universities and programs are being charged with interest only in bigness and its often alleged counterpart, anonymity.

Second, principles being taught in marketing, production and finance are reinforced through specific related assignments made in conjunction with the Simulation course. Such assignments seem far more relevant and urgent to the student than the typical potpourri of problems at the end of the chapter in textbooks.

Third, the integration and assimilation by each student of knowledge in marketing, production, and finance into a management philosophy is fostered. Since each team is required in the Simulation course to make several marketing, production and financial decisions over an extended period of simulated time, it must do so from an interrelated point of view if it is to operate effectively. The learning of that delicate quality called "balance" is fostered.

Fourth, integration is achieved not only with respect to the student, but the faculty, also. For the integrative core to be successful, there must be serious planning and coordination of individual course topics, assignments and examinations. As a result, duplication of subject matter is eliminated and more importantly, gaps are closed. Overloads are coordinated. By precept and example, faculty show their acknowledgement of and respect for all functional areas, regardless of their specialities, and particular research interests.

Implementation Details

The implementation details of the integrative core will be discussed primarily from the point of view of the Simulation course. As a matter of review, a business simulation or game may be defined as a sequential decision-making exercise structured around a model of business operation, in which participants assume the role of managing the simulated operations[2]. Business simulations abound today, but few are both sufficiently complex and comprehensive to provide the basis for a one semester, three credit hour course. One that does meet both criteria is INTOP (International Operations Simulation of the University of Chicago) developed by Rorelli and Graves[6,7,8]. Of course, the purpose of this paper is not to report about INTOP; however, the reader must have some conception of the character of INTOP, if he is to appreciate the role that the Simulation course is capable of performing in the integrative core. The "typical" business simulation can be represented in a 1000-2000 card FORTRAN program. The INTOP model is represented in approximately 9000 FORTRAN cards. If one makes the dangerous assumption that there is a direct relationship between the number of cards and game complexity and comprehensiveness (and realism), INTOP and the few models like it are in a separate league. INTOP is not a program to be "dumped-on" the small departmental computer one day and processed the next day. INTOP is international in scope and allows fundamental decision-making in all functional areas of business except production balancing and raw materials purchasing. Environmental parameters may be changed so as to emulate segments of recent international economic activity. Typical computer output in the form of financial statements, marketing reports and ancillary data is shown in Illustration 1.

Procedurally, the Simulation course meets in 75 minute sessions twice a week for 15 weeks. Individual sections are limited to nine teams. The first few class sessions are used to explain the purposes and mechanics of the course and then the course is divided into two separate, but related parts, occupying the two weekly class sessions.

One session can be described as open company meetings; that is, teams (companies) are in open session analyzing operations and results, formulating plans and otherwise getting ready for their new round of management decisions which are made on a weekly basis (and represent one

ILLUSTRATION 1. SAMPLE INTOP OUTPUT.
COMPANY 1

INTOP - UNIVERSITY OF CHICAGO

PERIOD 6

INCOME STATEMENT	AREA 1		AREA 2		AREA 3		HOME OFFICE CONSOLIDATED
	PRODUCT X	PRODUCT Y	PRODUCT X	PRODUCT Y	PRODUCT X	PRODUCT Y	
STANDARD SALES							
CONSUMER	72000	1131561	491328	1248480	313791	0	3913160
INTRA-COMPANY	0	0	0	0	0	0	0
INDUSTRIAL	0	0	0	0	0	0	0
LESS-COST OF GOODS	227546	324718	134309	297072	13517	0	997162
GROSS MARGIN	500454	806843	357018	951408	300274	0	2915997
DELUXE SALES							
CONSUMER	0	0	0	0	0	0	0
INTRA-COMPANY	0	0	0	0	0	0	0
INDUSTRIAL	0	0	0	0	0	0	0
LESS-COST OF GOODS	0	0	0	0	0	0	0
GROSS MARGIN	0	0	0	0	0	0	0
TOTAL GROSS MARGIN	500454	806843	357018	951408	300274	0	2915997
OPERATING EXPENSES							
COMM. AND ADMIN.	110861	112006	74298	109736	56862	0	661762
ADVERTISING	53000	78000	40000	51000	23000	0	245000
SHIPPING	48000	0	0	0	0	0	48000
INVENTORY	0	34057	1498	3437	0	0	38992
SALES EXPEDITING	0	0	0	0	0	0	0
METHODS IMPROVEMENT	10000	15000	5000	8000	0	0	38000
DEPRECIATION AND FIXE	215000	190000	100000	145000	0	0	650000
NET OPERATING EXPENSE	436861	429063	220795	317173	79862	0	1483754
NET EARNINGS FROM OPER.	63592	377780	136223	634235	220413	-0	1432243
TOTAL NET OPER. EARNINGS		441372		770458		220413	1432243
NON-OPERATING INCOME							
INTEREST INTERCO. LOANS						0	0
LICENSES-X						0	0
LICENSES-Y						0	0
MISC. INTEREST		1000		1000		0	14750
TOTAL NON-OPER. INCOME		1000		1000		0	14750
NON-OPERATING EXPENSE							
MARKET RESEARCH						72000	72000
LICENSES-X						0	0
LICENSES-Y						0	0
R AND D NEW PRODUCT X						0	0
R AND D NEW PRODUCT Y						0	0
TOTAL INTEREST		0		0		0	99000
TOTAL NON-OPER. EXPENSE		0		0		0	171000
GROSS EARNINGS		442372		771458		220413	-158250
LESS-TAXES		230034		347156		66124	0
LESS-CAPITAL TRANS. TAX		0		0		0	0
NET EARNINGS		212339		424302		154289	-158250
LESS-DIVIDENDS						0	0
TD RETAINED EARNINGS		212339		424302		154289	-158250

BALANCE SHEET	AREA 1		AREA 2		AREA 3		HOME OFFICE CONSOLIDATED	
ASSETS								
CASH		12757		260351		106257	110793	490158
A/R FIRST QUARTER		1115737		476189		143379		1735305
A/R SECOND QUARTER		0		521942		125517		647459
INVENTORY STANDARD X	260630		116060		0			
DELUXE X	0		0		0			
STANDARD Y	191818		17018		0			
DELUXE Y	664982		353492		0			
TOTAL		1117431		486570		0		1604002
SECURITIES		100000		100000		0	750000	950000
TOTAL CURRENT ASSETS		2345925		1845053		375152	860793	5426923
NET PLANT AND EQUIP.		5160000		5000000		0		10160000
INVESTMENT INTERCOMP.							0	0
SUBSIDIARY CONTROL							11895000	
TOTAL ASSETS		7505925		6845053		375152	12755793	15586423
LIABILITIES								
A/P FIRST QUARTER		785401		477078		66124		1328603
A/P SECOND QUARTER		0		137869		0		137869
SUPPLIER CREDIT		0		0		0	0	0
AREA BANK LOANS		0		0		0	0	0
TOTAL CURRENT LIABILITY		785401		614947		66124	0	1466472
LOANS PAYABLE							3300000	3300000
TOTAL LIABILITIES		785401		614947		66124	3300000	4766472
STOCKHOLDER EQUITY								
COMMON STOCK AT PAR							10000000	10000000
PAID IN CAPITAL							0	0
RETAINED EARNINGS		395524		785106		184028	-544207	820451
HOME OFFICE CONTROL		6325000		5445000		125000		
TOTAL EQUITY		6720524		6230106		309028	9455793	10820451
TOTAL LIAB. AND EQUITY		7505925		6845053		375152	12755793	15586423

ILLUSTRATION 1, CONTINUED.
COMPANY 1

INTOP - UNIVERSITY OF CHICAGO

PERIOD 6

ANCILLARY DATA	AREA 1		AREA 2		AREA 3	
	PRODUCT X	PRODUCT Y	PRODUCT X	PRODUCT Y	PRODUCT X	PRODUCT Y
STANDARD SALES UNITS						
CONSUMER	28000	18859	15849	20808	8965	0
INTRA-COMPANY	8000	0	0	0	0	0
INDUSTRIAL	0	0	0	0	0	0
DELUXE SALES UNITS						
CONSUMER	0	0	0	0	0	0
INTRA-COMPANY	0	0	0	0	0	0
INDUSTRIAL	0	0	0	0	0	0
MFG. COST ANALYSIS						
PL(1) STANDARD COST	117351	0	106071	0	0	0
UNITS	18000	0	13000	0	0	0
DELUXE COST	0	664982	0	353492	0	0
UNITS	0	32000	0	21000	0	0
PL(2) STANDARD COST	143280	0	0	0	0	0
UNITS	20000	0	0	0	0	0
DELUXE COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
PL(3) STANDARD COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
DELUXE COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
STANDARD GRADE	0	0	0	0	0	0
DELUXE GRADE	0	1	0	1	0	0
INTRA-CO. PURCHASES						
STANDARD COST	0	0	0	0	0	0
UNITS	0	0	0	0	8000	0
DELUXE COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
INDUSTRIAL PURCHASES						
STANDARD COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
DELUXE COST	0	0	0	0	0	0
UNITS	0	0	0	0	0	0
ENDING INVENTORY						
STANDARD UNITS	38000	11141	14179	1192	8000	0
GRADE	0	0	0	0	0	0
DELUXE UNITS	0	32000	0	21000	0	0
GRADE	0	1	0	1	0	0
NO. REG. SALES OFFICES	1				1	
MAX. GRADE OF IMPROVEMENT	0	1				

MARKET RESEARCH 1	AREA 1				AREA 2				AREA 3			
	PRODUCT X		PRODUCT Y		PRODUCT X		PRODUCT Y		PRODUCT X		PRODUCT Y	
PRICES POSTED THIS QRT.	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.
COMPANY NUMBER 1	26	-0	60	-0	31	-0	60	-0	35	-0	-0	-0
COMPANY NUMBER 2	32	-0	56	-0	28	-0	50	-0	-0	-0	-0	-0
COMPANY NUMBER 3	28	-0	-0	-0	30	-0	60	-0	37	-0	-0	-0
COMPANY NUMBER 4	-0	-0	55	65	-0	-0	80	-0	-0	-0	-0	-0
COMPANY NUMBER 5	24	-0	44	57	-0	-0	-0	-0	-0	-0	-0	-0
COMPANY NUMBER 6	40	-0	70	85	-0	-0	75	89	-0	-0	-0	-0
COMPANY NUMBER 7	30	-0	65	-0	35	-0	57	82	-0	-0	-0	-0
COMPANY NUMBER 8	20	-0	85	-0	22	60	85	-0	-0	-0	-0	-0
COMPANY NUMBER 9	23	-0	52	-0	30	-0	54	-0	-0	-0	-0	-0
GRADES MFG. FOR NEXT QRT.												
COMPANY NUMBER 1	0	-0	-0	1	0	-0	-0	1	-0	-0	-0	-0
COMPANY NUMBER 2	0	1	0	-0	0	-0	0	-0	-0	-0	-0	-0
COMPANY NUMBER 3	0	-0	0	-0	0	-0	0	-0	-0	-0	-0	-0
COMPANY NUMBER 4	-0	-0	1	2	-0	-0	2	-0	-0	-0	-0	-0
COMPANY NUMBER 5	-0	1	0	1	1	-0	-0	-0	-0	-0	-0	-0
COMPANY NUMBER 6	0	-0	0	1	-0	-0	0	1	-0	-0	-0	-0
COMPANY NUMBER 7	0	1	1	-0	-0	-0	1	2	-0	-0	-0	-0
COMPANY NUMBER 8	-0	-0	1	-0	1	2	1	-0	-0	-0	-0	-0
COMPANY NUMBER 9	0	1	0	-0	-0	1	0	-0	-0	-0	-0	-0

NOTE: -0 DEMOTES NO PRODUCTION, 0 DEMOTES THAT ZERO IS THE GRADE BEING MANUFACTURED

MARKET RESEARCH 2	AREA 1				AREA 2				AREA 3			
	PRODUCT X		PRODUCT Y		PRODUCT X		PRODUCT Y		PRODUCT X		PRODUCT Y	
SALES THIS QUARTER(000)	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.	STD.	DEL.
COMPANY NUMBER 1	28.0	0.0	18.9	0.0	15.8	0.0	20.8	0.0	9.0	0.0	0.0	0.0
COMPANY NUMBER 2	10.6	0.0	22.2	0.0	17.0	0.0	28.0	0.0	0.0	0.0	0.0	0.0
COMPANY NUMBER 3	17.1	0.0	0.0	0.0	16.7	0.0	20.0	0.0	7.1	0.0	0.0	0.0
COMPANY NUMBER 4	0.0	0.0	7.6	37.5	0.0	0.0	19.9	0.0	0.0	0.0	0.0	0.0
COMPANY NUMBER 5	34.9	0.0	35.7	24.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
COMPANY NUMBER 6	.9	0.0	6.7	6.0	0.0	0.0	4.6	6.5	0.0	0.0	0.0	0.0
COMPANY NUMBER 7	15.9	0.0	12.3	0.0	7.6	0.0	22.9	5.4	0.0	0.0	0.0	0.0
COMPANY NUMBER 8	1.3	0.0	5.4	0.0	18.2	0.0	8.2	0.0	0.0	0.0	0.0	0.0
COMPANY NUMBER 9	35.9	0.0	28.5	0.0	14.9	0.0	26.0	0.0	0.0	0.0	0.0	0.0
TOTAL SALES(000)	144.6	0.0	137.3	67.8	90.2	0.0	150.3	11.9	16.1	0.0	0.0	0.0

quarter of business operations). The role of the instructor is to visit each team and, aided by managerial accounting data and statistics (to be explained later) from all previous quarters of company operations, to prod and question each regarding analyses, strategies, tactics and specific decisions for reaching previously established company objectives. The instructor, of course, aids companies having specific questions and/or any difficulties with analyses they should be able to do at any given time during the semester. The open class session also provides a convenient time for companies to negotiate all types of inter-company deals including industrial product sales, product licensing, leasing of facilities and services, etc. INTOP has sufficient flexibility to accommodate practically any intercompany transaction imaginable.

The other weekly session is more traditional, being devoted to instruction. The subject matter includes planning, controlling and the decision-making process. Considerable time is devoted to bringing the functional areas of management into focus, and integrating them with the operational decision-making taught in marketing, production and finance. A skeletal course outline is shown in Illustration 2. The topical presentation order in Welsch[9] is extremely complementary to the development of the lecture schedule and the simulation.

The first two to three weeks of the course are involved with explanation of the INTOP environment, learning of the specific rules and development of philosophy, objectives and organizational structure by each company. The fourth week the first round of decisions is due and then one round of decisions is due each week thereafter until a total of 12 quarters (3 years) of operations have been simulated. (Actually only 10 sets of decisions are completed; the first set of decisions is for three quarters of operations. It takes two quarters to build production facilities and one quarter to manufacture products, so product market interaction is not begun until the fourth quarter.)

Commencing with the first lecture on planning and control, a second computer program is processed with INTOP. The introduction of this summary analysis[5] is unique to gaming and has so enhanced the course that it warrants specific discussion. The program is run in conjunction with the INTOP program and subsequent to it; sample output is shown in Illustration 3. The instructor gets a copy of the summary analysis for all teams and each team receives a copy pertaining to its operations, minus the summary rankings; these are on the instructor's copy only. The program solves two problems. First, in order for companies to make the most intelligible, rational, objective INTOP decisions, it is necessary for them to generate a mass of managerial accounting data, in addition to the ancillary data that is a part of the regular INTOP output. The generation of these data required students to engage in busy work that (1) did not enhance their learning and (2), in fact, limited even further the amount of time available for true decision-making. In addition, the logical argument was forwarded that in the "real" world, managerial data would be systematically provided on a timely basis by the accounting department of the company. Note the wealth of detailed sales, production, inventory, financial and cost accounting information arranged by geographical area in the summary analysis. This type of information measurably enhances managerial decision-making. The second problem was concerned with the systematic and periodic analysis of each team's progress and problems. The amount of INTOP output generated weekly by a nine team section of Simulation of Business is voluminous. In order for the instructor to provide a meaningful critique for each team during the open session, the analysis of the output by hand was just too burdensome. The output format of the summary analysis is so arranged that the instructor can easily trace performance of a team in any particular aspect of the simulation by just reading across the appropriate page and line(s). Teams are required to submit forecasts by geographical area of sales, cash, production costs and ROI (return on investment). The weekly forecasts, first described as a written assignment for week five in Illustration 2, are required for every quarter of simulated operations. Any resulting high forecast errors tend to flag, automatically, problems that a team is having with particular aspects of the simulation. Armed with summary statistics, forecast errors, relative rankings and a complete historical summary of each company's operations, the instructor efficiently prepares for his conference with each team during the open session. Problem areas are quickly uncovered so that therapy may begin.

A secondary benefit resulting from requiring each team to make quarterly forecasts should be mentioned. The benefit is that guessing is practically eliminated. If a particular forecast of a team is in serious error, the first thing the instructor asks the team during the open session is to review with him the particular analysis that produced the forecast. Having none

ILLUSTRATION 2. Business Simulation Course INTOP Related Lecture Schedule

<u>Week</u>	<u>Topics</u>	<u>Assignment</u>
1-2	Business Philosophy and Objectives	Company Philosophy and Objectives
3	Organizational Structure	Company Organizational Chart and Individual Job Descriptions
4	Development of Specific Goals and Strategies	Company Goals and Strategies
5	Sales Planning; and Control; Forecasting	Forecast of Sales, Variable Costs, Net Earnings, Cash and ROI for each Quarter of Operations
6	Production and Inventory Planning	
7	Planning Expenses-Manufacturing Overhead, Distribution and Administrative Expenses	
8	Planning and Controlling Cash, Dividend Policy	
9	Planning and Controlling Capital Expenditures	Formal Proposal for a Major Capital Expenditure
10	Development of the Annual Profit Plan	Annual Profit Plan for Next Year of Operations
11	Cost-Volume-Profit Analysis	Cost-Volume-Profit Analysis Supporting an INTOP Decision
12	Variance Analysis	Variance Analysis of One Marketing Area
13	Demand Analysis and Price Setting	Demand Analysis for One Product in One Market Area
14-15	Company Audits	Oral Presentation of Audits

proves to be very embarrassing. Also, in order to do an acceptable job of forecasting, teams must engage in a systematic analysis of the variables that impact on the particular variable to be forecast.

The topics covered in weeks six through twelve are traditional and should need no further explanation here. Of course, the point to reiterate is that each of these topics is supplementary to the simulation and is presented as just another aid to improve the decision-making that each company is required to do.

The topic of demand analysis in week thirteen does warrant specific discussion. I think many business students, as a result of their classroom training, are convinced that the concept of a demand curve is one of those theoretical constructs which does not really have any practical value in aiding business decision-making. By week thirteen teams have generated sufficient data to plot a demand curve which is very aesthetically appealing in terms of fit. It is a real joy to watch students rediscover (or discover) one of the basic principles of economics and to determine that the principle is applicable to the simulated world in which they have been operating (which by this time in the semester is as real to the students as if they were running GM or IBM).

The last topic, company audits, also deserves comment. At the completion of three years of simulated operations, each company's records are turned over to an auditor (which is, in reality, one of the other companies). Records include, in addition to the standard financial reports produced as part of the INTOP output, required charts, graphs and brief narratives of salient discussion points covered prior to each decision. The auditing company is asked to do a complete audit in terms of management, finance, production, etc., culminating in a 15-20 minute oral presentation. Given the benefit of hindsight and a semester's practice at decision-making, an auditing team quickly flags the "boners" and the quality decision-making of the audited company. To insure that students never forget for a moment the need to plan for the future, each auditing team is asked to make a minimum of three specific major recommendations to the future management of the audited company.

Grading

The team being the basic decision unit (rather than the individual) is reinforced in the grading schema for the Simulation course; overall team performance accounts for 40 percent of the course grade. The paramount danger here, as with most simulations, is to reward intuitive behavior too generously. The rewards should go to the teams mastering sound management principles, prescribed analyses, etc. But, on the other hand, those teams that practice these principles and analyses and combine them with intuition, yielding a superior strategy, should also be rewarded. I will not argue further as to whether it is "how you play the game" or "whether you win or lose" that is important in terms of learning. Personally, I mix the two in equal proportion. One-half of the 40 percent is based upon INTOP written assignments - development of team philosophy, objectives, strategies, goals, analyses, sound forecasting methods, etc. Each visitation during the open session has been institutionalized via a simple summary performance sheet on which the instructor evaluates the quality of decision-making during the last quarter of operations. The other one-half of the team grade is based upon team performance vis-a-vis other teams. This is a particularly difficult grade to quantify because (1) it is fraught with all the perils that real companies have in quantitatively measuring performance and (2) the instructor needs to maintain consistency from one semester to the next. For example, does one utilize return on sales, return on investment or return on total assets as the quantitative measure of performance[1]? Also, how does one guard against the possibility of the "last place" team this semester being "better" than the "first place" team the semester before? If the same aggregate profit potential were generated each semester by the simulated environment, then objective measures of level of performance could be ascertained after a couple semesters. In INTOP, the economic environment is changed every semester to (1) prevent previous class notes on INTOP being of any value and (2) to emulate a recent period of economic activity to gain even more realism. Each environment has a different aggregate profit potential that is really not measurable, a priori. In an attempt to negate some of these problems, I use the following grading procedure for this portion of team performance:

1. Rank teams according to the sum of their retained earnings and paid in capital as of period 12. This is a measure of each team's leadership position in the industry, based upon results to date.
2. Rank teams according to their average return on capital over quarters 4-12. This is a measure of each team's efficiency in utilization of capital, regardless of source or how much they use.
3. Rank teams in their average plant and equipment investment over quarters 8-11. This is a measure of each team's potential ability to do competitive battle in the future.
4. Take the 80th percentile team (in terms of number of teams participating) and assign the team with that ranking a 100 percent rating. Calculate the

ILLUSTRATION 3. SAMPLE INTOP ANALYSIS REPORT - INDIANA UNIVERSITY SCHOOL OF BUSINESS - COMPANY 2.

	QTR 1	QTR 2	QTR 3	QTR 4	QTR 5	QTR 6
***** AREA 1 (US)						
ECON INDEX X				1.06	1.07	1.02
SALES X STD-GRD				0	0	0
CONSUMER PRICE				18203	16985	10572
OTHER PRICE				24	25	32
TOTAL FORECAST				0	0	6000
PCT ERROR				0	0	16572
SALES X DLX-GRD				18203	16985	16572
CONSUMER PRICE				16000	15000	19000
FORECAST				13.77	13.23	-12.78
PCT ERROR						1
ECON INDEX Y				.93	.97	.91
SALES Y STD-GRD				0	0	0
CONSUMER PRICE				31000*	30508	28908
FORECAST				43	45	44
PCT ERROR				29000	29000	29500
				6.90	5.20	8.16
ADVERTISING						
PRODUCT X				21000	21000	21000
PRODUCT Y				31000	30000	29000
SALES OFFICES						
C AND A COSTS				0	0	2
X ACTUAL				3.30	3.30	3.30
X OPTIMAL				2.85	2.85	2.85
Y ACTUAL				4.00	4.00	4.00
Y OPTIMAL				4.27	4.27	4.00
AVG CUG SOLD						
X STD-GRD				6.28-0	6.04-0	7.07-0
X DLX-GRD				0.00-0	0.00-0	8.73-1
Y STD-GRD				16.68-0	16.79-0	18.18-0
METHOD IMPROVEMENT						
PRODUCT X				15000	15000	16000
PRODUCT Y				12000	12000	13000
PROD COSTS						
X PLANT 1-AGE	1	2	3	4	5	6
QTY-GRD			19000-0	19000-0	21000-0	18000-0
AVC			6.28	6.04	7.07	6.31
FORECAST			8.50	6.35	6.50	6.16
PCT ERROR			-26.12	-4.88	8.77	2.60
X PLANT 2-AGE			1	2	3	4
QTY-GRD					19000-1	19000-1
AVC					8.73	9.13
FORECAST					9.50	8.70
PCT ERROR					-8.10	4.94
Y PLANT 1-AGE	1	2	3	4	5	6
QTY-GRD			31000-0	31000-0	35000-0	30000-0
AVC			16.68	16.79	19.58	17.59
FORECAST			18.50	16.70	17.50	17.00
PCT ERROR			-9.83	.01	11.89	3.47
INVENTORY CARRIED FORWARD						
X STD-GRD				797	2812	7240
X DLX-GRD						12574
X CUST/UNIT				.53	.89	1.32
Y STD-GRD				0	492	3584
Y CUST/UNIT					2.42	2.71

ILLUSTRATION 3. CONTINUED

ENDING INVENTORY							
X STD-GRD	19000-0	19797-0	23812-0	25240-0			
X DLX-GRD	0-0	0-0	19000-1	31000-1			
Y STD-GRD	31000-0	31000-0	35492-0	33584-0			
CASH	-112359	666395	431388	75640			
FORECAST	200000	200000	200000	350000			
PCT ERROR	-156.18	233.20	115.69	-78.29			
SECURITIES			700000	500000			
NET EARNINGS	-320000	33874	347326	542100			
NET EARN X	-130000	107960	131477	-36846			
FORECAST X	-130000	53000	98560	200000			
PCT ERROR	0	103.70	33.40	-118.42			
NET EARN Y	-100000	455945	514547	460905			
FORECAST	-100000	302000	450000	470000			
PCT ERROR	0	50.98	14.34	-19.25			
CURRENT RATIO	2.65	9.16	3.68	4.10			
ROI	-10.67	9.16	5.00	2.85			
***** AREA 2 (FEC) DELETED REPORT							
***** AREA 3 (BRAZIL) DELETED REPORT							
***** HOME OFFICE PARTIAL REPORT							
MKT RESEARCH							
ITEM 1	1	2	17				
ITEM 2		5					
ITEM 3							
MAXIMUM PRODUCEABLE GRADE							
PRODUCT X	0	0	1	1	1	2	
PRODUCT Y	0	1	1	1	2	2	
R&D PRD X	60000	60000	60000	40000	80000	80000	
R&D PRD Y	120000	120000	120000	80000	40000	40000	
DIVIDENDS							
NET EARNINGS (CONSOL)	0	-215161	-957688	-348990	395942	857564	
ROI (CONSOL)	-1.07	-1.09	-7.58	5.50	7.38	3.87	
FORECAST	-3.00	-2.05	-0.40	3.90	6.10	6.05	
PCT ERROR	64.33	46.83	19.36	41.03	20.98	-36.03	
AVG FORECAST ERROR							
UNIT SALES				12.56	10.85	-5.98	
UNIT VAR COST				-4.35	-6.10	2.46	
NET EARN OPS				40.32	19.85	-15.60	
CASH	171.25	-160.38	-153.53	90.65	75.49	-60.10	

EFFICIENCY MEASURE SUMMARY - QUARTER 6

TEAM	CASH/SALES		SUP CRP/SALES		INVT COST/SALES		NET EARN/SALES		ROI	
	O/D	RANK	O/D	RANK	O/D	RANK	O/D	RANK	O/D	RANK
1	11.57	3	.63	3	.46	2	29.04	7	8.85	3
2	18.69	7	0	1	3.72	8	32.18	6	3.87	9
3	3.85	1	.34	2	1.51	7	23.04	9	4.86	7
4	26.00	9	0	1	.18	1	41.56	2	8.05	6
5	24.68	8	0	1	1.31	6	35.54	5	9.90	2
6	4.83	2	6.14	4	4.58	9	23.53	8	4.76	8
7	17.38	5	0	1	.62	3	48.50	1	12.71	1
8	17.67	6	0	1	.90	5	39.67	3	8.85	4
9	16.38	4	0	1	.91	4	36.54	4	8.62	5

ILLUSTRATION 4. Sample Peer Analysis Form

CONFIDENTIAL

GROUP ANALYSIS

Your Name _____ Company (#) _____ (Name) _____

You are assured that all data on this sheet will be held in strict confidence.

List each member of your company, including yourself. For each of the characteristics listed, rate each member including yourself, on a 4.0 scale (2.0 = C)

Names in Slanted Columns

	1	2	3	4	5
1. Attendance at Team and functional meetings					
2. Completion of tasks delegated to individual					
3. Quality of work done by individual on delegated tasks					
4. Frequency of ideas					
5. Value of ideas					
6. Ability to plan ahead					
7. Use of analytic techniques					
8. Ability to see and analyze other points of view					
9. Carrying fair share of company's work load					
10. Overall value to company					

percentage rating of each team in each performance criterion by dividing the amount attained in each by the amount attained by the base team.

5. Calculate a composite percentage for each team and grade according to a pre-established basis such as 90% = A, 80% = B, 70% = C, etc.

An additional 25 percent of the total grade is allocated to individual performance of team members. Except by subjective evaluation, it is difficult for the instructor to determine whether each member of a team is carrying his "fair share" of the load. Therefore, a portion of the total grade is allocated to individual performance of team members, and it is basically determined by the students, themselves. Utilizing a standardized grading form (see Illustration 4) provided at the beginning of the semester, each team member grades all team members on ten aspects of individual performance at the end of the semester. This forces students to assess one another as working members of a small group. Students have accepted this grading responsibility quite seriously, rewarding and penalizing according to contribution or lack thereof, respectively. The remaining 35 percent of the grade is allocated to traditional quizzes and examinations over course material.

Results - Discussion

At the 1971 Conference on Computers in the Undergraduate Curricula, Meyers[3] presented a paper entitled, "We Don't Know What We Are Doing." His basic theme was that we need to measure more objectively the gains, if any, resulting from the use of computers in the undergraduate curricula. I heartily agree with his theme and am sorry to note that we have not done any systematic objective measurement of the gains (or losses), especially when related to costs resulting from the introduction of the integrative core. Therefore, I think it important to note the sub-heading title, Results - Discussion, instead of the more popular title, Conclusion.

The integrative core at Indiana University is not "new" in terms of subject matter. Rather, it is the traditional business material wrapped in what we think is a more attractive package. The expressed interest of students has been very high when compared with the prior curriculum. The quality of the students' educational experience has been enhanced by the obvious integration of courses. The Simulation course literally forces the student to pull things together. The students know the four courses are interrelated by program design. Many of their classroom comments and questions are related to work from one or more of the other core courses. The content of the students' educational experience is enhanced by the presentation of traditional theory within a framework of practical application. Students see how such techniques as cost-volume-profit analysis, capital budgeting, "keep-out" pricing, etc. are utilized by a business firm. They learn about the usefulness of theory in an environment substantially of their own making. They also learn about the limitations and difficulties of trying to apply theory when other things seldom remain so conveniently *ceteris paribus*. The advantage of continually forcing the transfer of theory into practice seems obvious to me.

"Poorer" students appear more motivated in the integrative core environment. Team activities provide the framework for additional learning outside the classroom. Students are willing to ask teammates about particular problem areas (and in the process implicitly admit their lack of understanding); whereas, they will often not ask the instructor for assistance. That overall understanding is increased and, at least, partially retained does not seem such a rash subjective conclusion to make.

At least two objective conclusions can be made. First, considerable duplication in the curriculum was eliminated. This resulted from faculty interaction aimed at development of the integrative core. It is interesting and sometimes embarrassing to find the same concept being taught in several different courses. Hopefully, the elimination of duplication has freed time for the introduction of other important material previously foregone. The second conclusion which may seem trivial in retrospect, but certainly was not in the beginning, is that it is physically possible to offer the integrative core concept. The obvious question is, "What are the costs?" Administrative, computer and faculty costs are involved. The administrative cost involved with strictly enforcing the prerequisites and arranging student and class schedules so as to provide team integrity varies according to the routine presently in the system but should not be overlooked. Computer costs for the Simulation course vary directly with the number of teams and the quarters to be simulated; however, a liberal estimate is two minutes of CDC 6500 central processor time for processing and analysis of one quarter of simulated operations for a maximum of 25 teams. Central memory requirements are 120,000 words of core for each program. Both programs are already overlaid extensively; therefore, the stated core requirements are minimal. Because most University computer centers operate "funny money" accounts, it is difficult to attach "real" dollar costs, but they are quite modest. The major cost is in terms of human resources. In order for the core to work, periodic coordination of individual course materials and development of integrative assignments must be accomplished. Due to the unusual amount of work involved in the Simulation course, the instructors are given double teaching load credit for each section of the course that they teach. In addition, two or three graduate

assistants for each set of 25 teams are needed to administer INTOP. One assistant is generally responsible for handling computer processing and the second is responsible for editing an industry organ that is a regular part of the output provided. In addition, the editor assists the instructors in evaluating team performance since many aggregate evaluations appear in the organ in one form or another. For the Simulation course, limited economies of scale exist with increasing numbers of students.

Student reaction has been extremely favorable. The one most often expressed dissatisfaction is that completion of the core demands an unrealistic amount of time. The amount of work and study demanded is extensive, but it is so by design. The only way to survive in the Simulation course is to divide and delegate the workload. There simply is not time for one student to carry the entire burden; thus, he is forced to put faith and confidence in the recommendations and analyses of fellow team members concerning their particular areas of operation. I have watched teams advance from last place (by practically any criterion of measurement) to first place simply as a result of a structural reorganization. What better way exists for teaching such concepts?

The Future

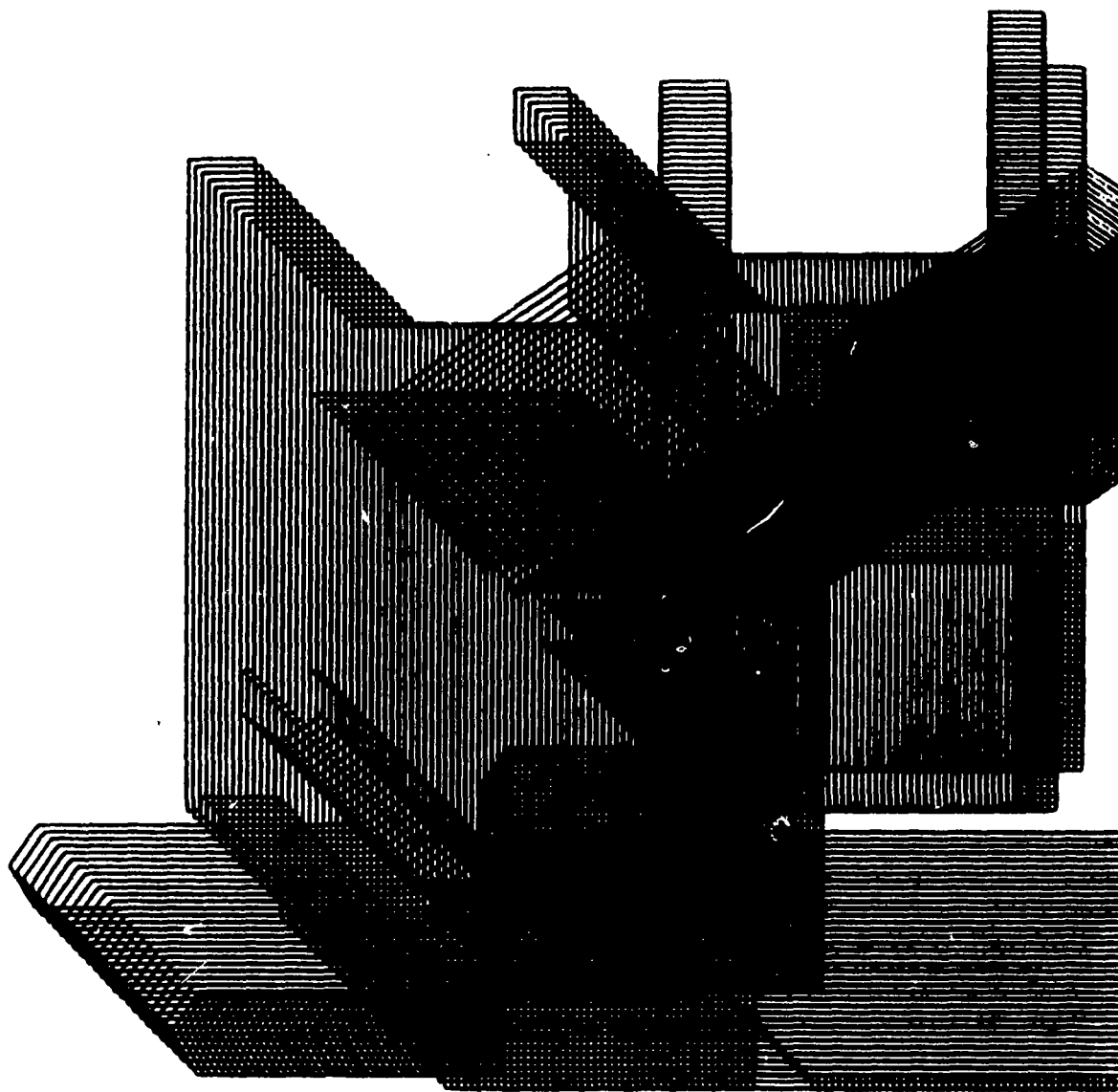
To date, the assessment by the faculty of the integrative core has been extremely favorable. However, the faculty believes that many potential advantages of the concept are yet to be exploited. The challenge is to create more meaningful assignments in each of the functional areas utilizing the simulation output. I believe there is practically no end to the integrative problems and assignments that may be developed.

ACKNOWLEDGEMENTS

The author cannot possibly emphasize enough the fact that the curriculum program described herein was conceived and developed by the faculty at the mother campus, Bloomington, and especially by Professor Panscher, chairman of the undergraduate program. Also, Professor Thorelli, INTOP's primary designer has provided much of the motivation for development of specific course materials. The author's experience with the undergraduate core program results from his responsibility for teaching the Simulation course at the Port Wayne Campus of Indiana University and from the excellent interaction which occurs among the other similarly responsible business professors at the seven campuses and himself. Some personal flairs described at the course presentation level and some of the reactions to the core are those of the author.

REFERENCES

1. Chamberlain, Neil W. The Firm: Micro-Economic Planning and Action, New York: McGraw-Hill Book Company, 1962, pp. 57-65.
2. Greenlaw, Paul S., L. W. Herron and R. W. Rawdon, Business Simulation in Industrial and University Education, Englewood Cliffs: Prentice-Hall, 1962.
3. Meyers, Edmund D., Jr., "We Don't Know What We Are Doing," Conference on Computers in the Undergraduate Curricula, Dartmouth College, 1971.
4. Panscher, William G., "The Four Course Integrative Core," Unpublished paper, 1969.
5. Smead, Raymond J., and Rodney W. Aldrich, Intop Analysis Program, Unpublished manual, Indiana School of Business, 1971.
6. Thorelli, Hans B., and Robert T. Graves, International Operations Simulation, New York: The Free Press, 1964.
7. Thorelli, Hans B., R. L. Graves and L. T. Howells, INTOP Player's Manual, New York: The Free Press, 1963.
8. Thorelli, Hans B., INTOP (2nd. edition), Indiana Readings in Business #76, Indiana University, 1972.
9. Welsch, Glenn A., Budgeting: Profit Planning and Control (3rd. edition), Englewood Cliffs: Prentice-Hall, 1971.



STRUCTURE by Steve Jaroch

Problem: Continuous line design, offset on the X, Y, and X/Y values

THE TIME-SHARING COMPUTER AND INTERMEDIATE ACCOUNTING

Elbert B. Greynolds, Jr.
Georgia State University
Atlanta, Georgia 30303
Telephone: (404) 658-2882

During 1970 at Georgia State University, we investigated the integration of a time-sharing computer system into selected undergraduate accounting courses. Since work of this nature had been performed in introductory accounting, accounting-systems and auditing, we selected the intermediate accounting and cost accounting courses for our study.

Most of the applications discussed in the accounting literature require "batch processing" computer systems as equipment. As an illustration, consider the excellent introductory accounting program library and text material developed by Wilbur P. Pillsbury, which he discussed at the 1971 conference at Dartmouth College[1]. His material was designed primarily for introductory accounting and a batch processing computer system. However, we wanted to work with advanced accounting courses and a time-sharing computer system.

We were anxious to exploit the flexibility offered by an interactive computer system. The concept of having the student use the computer as an analysis tool as well as a problem solver in accounting excited us.

Objective

Our objective was to integrate a time-sharing computer system in selected accounting courses. There were two qualifications in our objective. First, the method of integration should allow initial implementation in the courses without causing significant changes in their structures. Second, the computer should initially be a supplemental aid in the courses.

Scope and Limitations

Material was developed for cost accounting and intermediate accounting courses, but testing was restricted to the intermediate area. We chose the intermediate courses for several reasons. First, work of this nature is needed in the area since computer applications in intermediate are not as apparent as in cost accounting. Most of the cost applications are readily recognized as being valuable student aids.

The computer equipment used in the study was restricted to time-sharing computer systems with remote terminal capability. Excluded from use in the study were batch processing systems and programming languages such as FORTRAN and COBOL. Because we wanted to keep the project on an elementary level, the programming language BASIC was used for all programs. The simplicity of BASIC and its wide usage on other systems were key reasons for its selection.

All programs were developed and tested using the Georgia State RCA Spectra 70, Model 46, Time-Sharing System. Students had access to approximately twenty terminals during the study.

RESEARCH

Program Library

The first step in our study was obtaining a suitable time-sharing program library of applications. We investigated the time-sharing programs available to educational institutions. A review of these programs, however, revealed substantial duplication. Most of the programs were designed for applications in statistical, mathematical and other non-business areas. The financial programs available were oriented toward "Finance" courses and generally were not suitable for our study. When we narrowed the review to programs written in "BASIC," the deficit in financial accounting programs was apparent. As a result, we decided to develop the accounting time-sharing library at Georgia State.

Development of Program Library

The financial accounting library was developed using several guidelines. First, representative accounting topics were selected with as little duplication as possible of existing programs. Second, the programs were written so that students could use them as problem solvers or analysis aids. We wanted to release the student from the burden of lengthy

computational problems as well as allowing him the luxury of concentrating on the analysis of accounting problems rather than their solution.

The programs for the accounting library were evaluated according to six standards described below. The first four were primarily satisfied by proper program design, and classroom testing satisfied the last two standards.

First, the material for which programs are developed should be contained in current accounting textbooks. Second, the programs should be capable of utilization in connection with a representative number of problems contained in current accounting textbooks. As a result, a reasonable reduction in computation time should result. Third, program terminology should be consistent with accepted accounting terminology. Fourth, the programs should be referenced to specific accounting textbooks used in financial accounting courses. Change of a course textbook should require only minor changes, if any, in the program. Fifth, the programs should be easily applied by the students. Sixth, the programs should be reasonably free of errors.

Topic Selection

In selecting accounting topics for inclusion in the program library, we examined representative current accounting textbooks within the following classifications: Cost Accounting, Intermediate Accounting and Advanced Accounting.

Because the programs were to be tested at Georgia State University, emphasis was given to the textbooks used in the various accounting courses at State. The topics we selected for programming are also in other standard accounting textbooks. This selection does not exhaust all possible applications for program development; rather, the topics were chosen to illustrate the integration of a time-sharing computer system and accounting courses.

The accounting topics selected and programmed for the time-sharing library are:

1. Financial Statements
2. Financial Analysis
 - a. Ratio analysis
 - b. Break-even analysis
 - c. Cash budgets
 - d. Contribution margin analysis
 - e. Variance analysis
 1. Direct labor and material
 2. Overhead
3. Capital Budgeting
 - a. Present value computation
 - b. Internal rate of return computation
 - c. General purpose capital budgeting model
4. Compound Interest Calculations
5. Cost Accounting
 - a. Process costing
 - b. Reciprocal
6. Bank Reconciliation
7. Average Due Date of Account
8. Inventory Valuation
 - a. Lower of cost or market
 - b. Common inventory valuation methods
 - c. Retail inventory methods
9. Depreciation
 - a. Common depreciation methods
 - b. Annuity depreciation
 - c. Sinking fund depreciation
10. Coinsurance Calculations
11. Goodwill
12. Leasehold Amortization
13. Installment Loans
14. Pension Past Service Cost Amortization
15. Bond Analysis
 - a. Bond valuation and yield
 - b. Compound interest amortization of premium or discount
 - c. Gain or loss on bond redemption
 - d. Entries for purchases or sale of bond between interest dates
16. Serial Bonds
 - a. Serial bond valuation
 - b. Amortization of premium or discount
 1. Straight line
 2. Compound-interest
17. Sinking Funds
 - a. Ordinary fund
 - b. Fund for serial bonds
18. Stocks
 - a. Stock right valuation
 - b. Preferred and common stock dividends

The eighteen topics are contained in forty programs developed for the study. We also selected eight existing mathematical programs for inclusion in the library.

We wrote special instructions so the students could use the program with a minimum of assistance. The program instructions generally followed this format:

1. Program name
2. Purpose of program
3. References to specific accounting textbooks
4. Formats for data input
5. Sample problems contained in textbook

To facilitate student usage of the library, we compiled the program instructions in a manual: Time Sharing System Applications in Accounting(2). The manual also contains appropriate instructions for using the time-sharing system and terminals.

Library Evaluation

As previously noted, four of our six library evaluation criteria were satisfied by proper program and instruction design. The evaluation of the remaining criteria required student response, which was obtained by classroom testing. We basically wanted to find out if the programs were error free and if they could easily be used by students. We also sought their opinion on the value of the programs.

For our test, four Intermediate Accounting classes taught by three instructors were selected. This test was conducted during the 1970 Summer quarter. At Georgia State, Intermediate Accounting is a two quarter course. Our test group included three first quarter classes and one second quarter class. For evaluation purposes, we divided the classes into three groups--A, B and C. Group A, a first and second quarter class were given technical assistance in using the library. The other two classes, Groups B and C received no technical assistance. The computer background of the instructors varied. The instructors of Groups A and B had very little experience while Group C's instructor had extensive experience with commercial systems.

Group A's students were required to work a number of homework problems using the accounting library. The problems were selected from a list we furnished to the instructors.

The application of the programs in the other two groups was different. While student questions concerning program usage in Group A were answered, the other two groups used the programs on a "blind" basis. The instructors of Groups B and C received an explanation of the project and a list of the problems, after which no further assistance was given. The students in these groups used the programs on a voluntary basis.

The purpose of assisting them differently was two-fold. First, if the program instructions were sufficiently clear and the programs were generally free of errors, the students should be able to apply the programs without assistance. Second, it was desirable to sense or measure difference in response among the groups.

Questionnaire

A questionnaire was developed to obtain student opinion concerning ease of application, reasonable freedom from errors, and usefulness of the programs in problem solution. We made no attempt to measure either student understanding of accounting or knowledge of time-sharing systems. The purpose of the survey was to determine whether accounting students could use these programs in their courses. As a result, the evaluation is qualitative rather than quantitative.

The questionnaire given to the students at the end of the quarter contained thirty questions. The questionnaire results are summarized below, and we will supply details to interested parties.

The students' responses indicated that they could use the programs without difficulty and that the programs were generally error free. They indicated that using the computer in the accounting courses was beneficial to them, and that it should be continued. They also indicated that time was saved in working homework problems.

Because the programs were intended as a supplement to student learning and classroom teaching, the student must know the fundamental accounting techniques incorporated in each problem. This assumption was supported by the students. They indicated that the use of the program did not initially teach them the accounting techniques, but reinforced their understanding through analysis. The overwhelming majority of all groups indicated they would be willing to use the time-sharing computer in other courses and would be willing to learn BASIC.

The major complaint by the students concerned terminal availability. They felt that more terminals should be provided.

PROGRAM NAME--DEP8

PURPOSE:

THIS PROGRAM COMPUTES DEPRECIATION BY THE SINKING METHOD AND THE ANNUITY METHOD. FOR DETAILS SEE PAGES 636 TO 639 IN INTERMEDIATE ACCOUNTING BY WELSCH ZLATAKOVICH AND WHITE.

INSTRUCTIONS:

CALL UP THE PROGRAM AND TYPE RUN. DATA LINE STATEMENTS ARE NOT REQUIRED FOR THIS PROGRAM. THE INPUT VARIABLES ARE:

A-COST OF ASSET.
B-SALVAGE VALUE OF ASSET.
C-LIFE OF ASSET IN YEARS.
D-INTEREST RATE.

SAMPLE PROBLEM:

THIS PROBLEM IS TAKEN FROM THE TEXT QUOTED ABOVE ON PAGE 637.

COST = \$100
SCRAP VALUE = \$10
YEARS OF LIFE = 3
INTEREST RATE = 5%

OLD
OLD PROGRAM NAME--DEP8
READY
*RUN
ENTER COST (A), SALVAGE (B), YEARS(C) AND INTEREST RATE (D). A,B,C,D?100,10,3,.05

BOOK VALUE YEAR 0 = 100
SALVAGE IS 10
INTEREST RATE IS .05
LIFE IS 3 YEARS

DO YOU WANT ANNUITY PRINTED; YES (1), NO (0)
?1

***ANNUITY METHOD OF DEPRECIATION ***

PERIODS	INTEREST ON INVESTMENT	ANNUAL DEP. EXPENSE	CREDIT TO ACCUMULATED DEP.	BOOK VALUE OF ASSET
1	5	33.5493	28.5493	71.4508
2	3.57254	33.5493	29.9767	41.4741
3	2.0737	33.5493	31.4755	9.99854
TOTAL	10.6462	100.648	90.0015	

DO YOU WANT SINKING FUND METHOD; YES(1),NO(0)
?1

FIGURE 1. Program Name DEP8

SINKING FUND METHOD OF DEPRECIATION

PERIOD	ASSUMED DEPOSIT EACH PERIOD	INTEREST ON FUND BALANCE	DEP. EXPENSE	ACCUM. DEP.	BOOK VALUE OF ASSET
1	28.5493	0	28.5493	28.5493	71.4508
2	28.5493	1.42746	29.9767	58.526	41.474
3	28.5493	2.9263	31.4756	90.0015	9.99849
TOTAL	85.6478	4.35376	90.0015		

*

FIGURE 1. Continued

PROGRAM NAME--COINSUR

PURPOSE:

TO COMPUTE THE MAXIMUM INDEMNITY FOR A BLANKET POLICY OR WHEN A COINSURANCE CLAUSE IS INCLUDED IN A POLICY TO COMPUTE THE MAXIMUM INDEMNITY FOR ONE OR SEVERAL INSURERS. FOR A DISCUSSION OF THE COMPUTATIONAL DETAILS SEE INTERMEDIATE ACCOUNTING BY WELSCH, ZLATKOVICH AND WHITE, PAGES 366 TO 369.

INSTRUCTIONS:

DATA LINE INPUT IS REQUIRED FOR THIS PROGRAM. THE GENERAL FORMAT IS:

100 DATA N,P(1),P(2),...,P(N)
200 DATA N,R(1),R(2),...,R(N)

N-NUMBER OF POLICIES FOR BOTH DATA LINES.

P(1)-AMOUNT OF INSURANCE CARRIED (FACE VALUE) FOR EACH POLICY.

R(1)-CO-INSURANCE RATE FOR EACH POLICY IN LINE 100.

IF NONE OF THE POLICIES HAVE COINSURANCE CLAUSES, THEN DO NOT MAKE AN ENTRY IN DATA LINE 200, LEAVE IT BLANK.

IF THE POLICIES DO HAVE COINSURANCE CLAUSES, THEN A RATE MUST BE ENTERED FOR EACH POLICY EVEN IF ONE OF THE POLICIES DOES NOT HAVE A CLAUSE, IN SUCH A SITUATION ENTER A RATE OF 0.

USING THE FIRST EXAMPLE ON PAGES 368-69 OF THE TEXT, THE DATA LINES WOULD BE:

100 DATA 4,10000,25000,40000,5000
RUN

USING THE SECOND EXAMPLE ON PAGE 369, THE DATA LINES WOULD BE:

100 DATA 4,10000,25000,40000,5000
200 DATA 4,.9,.9,.9,.9

FIGURE 2. Program Name COINSUR

IF THE COINSURANCE RATES VARIED, THE DATA LINES WOULD BE:

100 DATA 4, 10000, 25000, 40000, 5000
 200 DATA 4, 0, .75, .8, .90

SAMPLE PROBLEM:

THE DATA FOR THIS ILLUSTRATION IS TAKEN FROM THE EXAMPLES FOR DATA LINE ENTRIES ABOVE.
 ALL USER SUPPLIED INFORMATION IS UNDERLINED.

OLD
 OLD PROGRAM NAME--COINSUR
 READY
100 DATA 4, 10000, 25000, 40000, 5000
 *RUN
 ENTER FAIR MARKET VALUE OF INSURED ASSETS (M), AND
 AMOUNT OF LOSS (L). M,L?
 ?100000,60000

CO-INSURANCE TABLE

POLICY NO.	CO-INSUR. CLAUSE	INSUR. CARRIED	REQUIRED BY CLAUSE	MAXIMUM COLLECTIBLE
1	0 %	10000	0	7500
2	0 %	25000	0	18750
3	0 %	40000	0	30000
4	0 %	5000	0	3750
TOTAL		80000		60000

100 DATA 4, 10000, 25000, 40000, 5000
 *200 DATA 4, .9, .9, .9, .9
 *RUN
 ENTER FAIR MARKET VALUE OF INSURED ASSETS (M), AND
 AMOUNT OF LOSS (L). M,L?
 ?100000,60000

CO-INSURANCE TABLE

POLICY NO.	CO-INSUR. CLAUSE	INSUR. CARRIED	REQUIRED BY CLAUSE	MAXIMUM COLLECTIBLE
1	90 %	10000	90000	6666.67
2	90 %	25000	90000	16666.7
3	90 %	40000	90000	26666.7
4	90 %	5000	90000	3333.34
TOTAL		80000		53333.4

FIGURE 2. Continued

PROGRAM NAME--PENSION

PURPOSE:

THIS PROGRAM COMPUTES THE ACCRUAL FACTOR, INTEREST AND TOTAL PENSION COSTS AS WELL AS THE CASH CONTRIBUTION AND BALANCE SHEET AMOUNT FOR PENSION PAST PERIODS SERVICE COSTS.

REFERENCE:

CONSULT INTERMEDIATE ACCOUNTING BY WELSCH, ZLATKOVICH AND WHITE PAGES 848 TO 890.

INSTRUCTIONS:

DATA LINE INPUT IS NOT REQUIRED FOR THIS PROGRAM. TYPE RUN AND ANSWER THE INPUT QUESTIONS. THE INPUT QUESTION VARIABLES ARE:

- A = THE INITIAL ACCRUED ACTUARIAL LIABILITY OR PAST SERVICE COST.
- B = THE NUMBER OF YEARS THE PAST SERVICE COST HAS BEEN ACCRUING.
- C = THE NUMBER OF YEARS THE PAST SERVICE COST WILL BE WRITTEN OFF OVER.
- D = THE INTERVAL OF YEARS FOR THE PRINT OUT. FOR EXAMPLE IF D = 5, THEN ONLY EVERY FIFTH YEAR WILL BE PRINTED IN THE TABLE
- R = THE INTEREST FACTOR FOR COMPUTING INTEREST.

SAMPLE PROBLEM:

8 YEARS AFTER ITS FORMATION, SMITH AND JONES CO. DECIDED TO ADOPT A PENSION PLAN. THE ACTUARY DETERMINED THAT THE INITIAL ACCRUED ACTUARIAL LIABILITY FOR PAST SERVICE COST AMOUNTED TO \$425,000. USING A 4% RATE AND ASSUMING THAT THE PAST SERVICE COST IS TO BE FUNDED IN 12 YEARS PREPARE A TABLE FOR DETERMINING THE ENTRIES NECESSARY TO RECORD THE PAST SERVICE COST. HAVE THE TABLE SHOW EVERY OTHER YEAR.

```
GLD
OLD PROGRAM NAME--PENSION
READY
*RUN
```

```
ENTER THE AMOUNT OF THE PAST SERVICE COST IN A.
ENTER THE NUMBER OF YEARS FOR THE ACCRUAL FACTOR IN B.
ENTER THE NUMBER OF YEARS IN THE WRITE OFF PERIOD IN C.
ENTER THE INTERVAL OF YEARS FOR PRINTOUT IN D.
ENTER THE EARNINGS RATE IN R AS A DECIMAL.
A,B,C,D,R
?425000,8,12,2,.04
```

FIGURE 3. Program Name PENSION

PENSION PAST PERIODS SERVICE COSTS

AMOUNT CHARGED TO PENSION EXPENSE

YEAR	ACCRUAL FACTOR	AMOUNT OF INTEREST	PENSION COST	CASH AMORT.	BALANCE SHEET AMT
1			63124.4	45284.7	17839.7
2	63124.4	713.587	63838	45284.7	36392.9
4	63124.4	2227.53	65351.9	45284.7	75755.4
6	63124.4	3865.01	66989.3	45284.7	118330
8	63124.4	5636.1	68760.5	45284.7	164378
10	0	5026.74	5026.74	45284.7	85410.5
12	0	1741.69	1741.69	45284.7	-847656

*

FIGURE 3. Continued

PROGRAM NAME--BDAMORT

PURPOSE:

THIS PROGRAM CALCULATES BOND AMORTIZATION BY THE SCIENTIFIC METHOD AS DISCUSSED IN INTERMEDIATE ACCOUNTING BY WELSCH, ZLATKOVICH AND WHITE PAGES 665 TO 669. FOR MORE INFORMATION CONSULT THE INTERMEDIATE TEXT.

INSTRUCTIONS:

CALL UP THE PROGRAM AND TYPE RUN. DATA LINE STATEMENTS ARE NOT REQUIRED FOR THIS PROGRAM. THE INPUT VARIABLES ARE:

- A-NOMINAL INTEREST RATE, FACE RATE.
- B-YIELD INTEREST RATE.
- C-TOTAL PAR VALUE OF BONDS IN DOLLARS.
- D-CARRYING VALUE OF BONDS, OR AMOUNT PAID FOR BONDS.
- E-NUMBER OF YEARS TO MATURITY FROM PURCHASE DATE.
- F-NUMBER OF MONTHS BETWEEN INTEREST PAYMENTS.

SAMPLE PROBLEM:

THIS PROBLEM IS FROM THE TEXT QUOTED ABOVE ON PAGE 665.

BONDS PURCHASED ON APRIL 1, 1970 WITH A PAR VALUE OF \$10,000. THE BONDS MATURE ON APRIL 1, 1973, AND THE NOMINAL INTEREST OF 5% IS PAID EVERY SIX MONTHS BEGINNING WITH OCTOBER 1, 1970. THE PURCHASE PRICE OF THE BONDS WAS \$10,280.07, AND THE YIELD WAS 4% ON AN ANNUAL BASIS.

@LD

@LD PROGRAM NAME--BDAMORT

READY

*RUN

NOMINAL INTEREST RATE (A), YIELD(B), TOTAL PAR OF BONDS (C), CARRYING VALUE OF BONDS (D), NUMBER OF YEARS TO MATURITY (E), MONTHS PER INTEREST PAYMENT (F). A,B,C,D,E,F
?.05, .04, 10000, 10280.07, 3, 6

FIGURE 4. Program Name BDAMORT

1968

79

BOND PREMIUM (DISCOUNT) AMORTIZED BY SCIENTIFIC METHOD

CONSTANT CASH PAYMENT (RECEIPT) 250
 ORIGINAL CARRYING VALUE OF BOND 10280.1

PERIODS	CASH RECEIPT (PAYMENT)	INTEREST INCOME (EXPENSE)	BOND PREMIUM (DISCOUNT)	INVESTMENT CARRYING VALUE
1	250	205.601	44.3987	10235.7
2	250	204.713	45.2868	10190.4
3	250	203.803	46.1925	10144.2
4	250	202.864	47.1163	10097.1
5	250	201.941	48.0587	10049
6	250	200.96	49.0199	10000
TOTALS	1500	1219.93	250.073	

*RUN
 NOMINAL INTEREST RATE (A), YIELD(B), TOTAL PAR OF BONDS (C), CARRYING VALUE OF BONDS (D), NUMBER OF YEARS TO MATURITY (E), MONTHS PER INTEREST PAYMENT (F). A,B,C,D,E,F
 ? .05, .06, 10000, 9729.14, 3, 6

BOND PREMIUM (DISCOUNT) AMORTIZED BY SCIENTIFIC METHOD

CONSTANT CASH PAYMENT (RECEIPT) 250
 ORIGINAL CARRYING VALUE OF BOND 9729.14

PERIODS	CASH RECEIPT (PAYMENT)	INTEREST INCOME (EXPENSE)	BOND PREMIUM (DISCOUNT)	INVESTMENT CARRYING VALUE
1	250	291.874	-41.874	9771.01
2	250	293.13	-43.1301	9814.14
3	250	294.424	-44.4241	9858.56
4	250	295.757	-45.7569	9904.32
5	250	297.13	-47.1294	9951.45
6	250	298.543	-48.5432	10000
TOTALS	1500	1770.86	-270.858	

*

FIGURE 4. Continued

At the close of the Summer quarter, the three test groups' professors discussed the student evaluation of the project. Their comments supported the responses of the students. They considered the program library to be a valuable tool in the classroom, easily applied, reasonably free of errors, and capable of utilization with current textbooks. They agreed that the programs were successfully introduced into their courses without causing significant changes.

No appreciable difference in student responses was found within the various groups. The three groups all had similar questionnaire results, even though there were differences in the computer experience of instructors and in the method of applying the programs.

Conclusions

After evaluation of student and teacher responses and the time-sharing library, we considered the objective of the study to be achieved. We successfully demonstrated the integration of a time-sharing computer system with intermediate accounting courses. We did this on a practical basis. Students and teachers are not required to possess programming experience, nor are significant changes in the courses required.

We are heartened by the students' response toward using the computer in their classes. This is an intangible result, which we believe to be important. They appeared to display increased interest in accounting computer applications, and apparently are more comfortable with the computer after using the library.

By using our approach, the computer can be introduced into intermediate accounting with minimal disruption. We believe that the same results as above could be obtained in cost accounting courses if the appropriate programs were tested and evaluated.

Some comments on the resources necessary for the development of a library similar to ours are appropriate at this point. The programs in our library were developed by us in six months on a part-time basis. We had full-time access to one teletypewriter terminal.

We will be happy to supply copies of the programs to interested individual users upon request. Just send us a magnetic tape and the appropriate specifications for your system. Copies of the instruction manual for the library are available from the Georgia State University Bookstore.

Program Examples

The program instructions were prepared using the computer. The instructions were stored in programs which allowed us to modify them when necessary. This also explains why all upper-case printing is used in the examples. The examples we selected for illustration are taken from the instruction manual. Because of the length of some instructions, we have selected short examples.

The accounting areas illustrated are:

- | | |
|------------------|--|
| 1. Depreciation: | DEP8--Sinking Fund and Annuity Depreciation |
| 2. Insurance: | COINSUR--Co-insurance Calculation |
| 3. Pensions | PENSION--Write-off of past service costs |
| 4. Bonds | BDAHORT--Bond amortization by the compound-interest method |

The text referred to in the instructions is Intermediate Accounting by Welsch, Slatkovich, and White, published by Richard D. Irwin. This was the intermediate text used by the school during the study.

REMOTE TIME-SHARING FOR EDUCATION IN BUSINESS PLANNING AND CONTROL

Harley M. Courtney
The University of Texas at Arlington
Arlington, Texas 76010
Telephone: (817) 273-3481

Introduction

Business planning and control courses have traditionally suffered from an inability to communicate the discipline to students in an adequate fashion. Typically the student is told how business firms perform the planning and control function, and some exercises are solved which illustrate the tools available to corporate planners. But the assigned exercises are usually so simple as to preclude any substantive understanding of the total management process, and consequently, are an inadequate representation of the merits and limitations of the tools utilized. Moreover, the simplistic nature of the exercises precludes the acquisition of any significant grasp of the behavioral aspects of the decision process or of the heuristic nature of business management.

But the appearance of time-shared computer terminal facilities on many university campuses promises the possibility of substantive courses in business planning and control. The facilities will free students from the computational constraints presently existing and will thereby permit the realistic illustration of mathematical models as well as the behavioral dimensions of the discipline.

In 1970, The University of Texas at Arlington offered a new course titled "Management Planning and Control," required of all accounting majors. The course has been popular with them and has attracted numerous students from other business disciplines, notably finance and management. Upon the initial offering of the course, attempts were made to utilize a batch-processing computer facility on campus. This step was of some benefit since complex cases having predetermined solution paths could be analyzed. But severe problems remained. Most planning and control problems did not appear to have solution paths which could be mapped in advance of the data analysis. Multiple runs of data were possible, but the turnaround time ranged from hours to days. This is probably typical of most campus computer centers and effectively precludes any significant number of multiple runs. When one remembers that designers of CAI systems feel that a response time of five seconds or less is necessary to maintain student interest, it is easy to understand why a batch system is inefficient.

Not only were case analyses difficult from the computational viewpoint, but the behavioral and heuristic dimensions of management choice could only be alluded to by the instructor rather than being illustrated and experienced by the student during the process of case analysis.

This latter problem is particularly acute in corporate planning and control courses since they are both tool-oriented and integrative, utilizing accounting and statistics on the one hand, and psychology and management organizational behavior on the other. In such an environment, computational inadequacies result in the student being told how a task should be accomplished, rather than being confronted with an illustration of the task and the opportunity to participate in a reasonable facsimile of it.

The availability of time-shared computing has permitted this highly desirable relief from computational constraints at UT-Arlington. During the summer of 1970 the first tentative steps were taken to integrate the use of time-shared remote terminals into the instruction of business planning and control. While the initial use was limited so as to determine its feasibility and instructional value, rapid growth in use has occurred. At present, all course instructors use the facility and its use has spread from one course topic to most of the topics considered in the course.

Applications of Time-Shared Computing

Initial applications of the time-shared terminal were relatively simple to implement. Nevertheless, the benefits were substantial. Two early uses were the solution of cost-volume-profit and capital budgeting cases by the use of canned programs authored by the instructor. These programs were interactive in nature such that students having no previous acquaintance with the computer were able to utilize them with no instructions other than the steps involved in signing-on, calling the programs, and signing-off.

Exhibit 1, for example, illustrates the use of a simple capital budgeting problem. All typing by the user in the exhibit is indented; all typing beginning at the left margin was under computer control. Thus the user typed the name of the program - CAPBUDGET - and the terminal response was to ask the user questions to which he responded by typing in the answer. Note that the student cannot only solve problems with unique inputs, but he can also perform a sensitivity analysis. The second example run in Exhibit 1 assumes the construction of a commercial building with relative certainty as to revenues since leases have already been signed, but uncertainty regarding the final construction cost. Thus several possible total costs can be entered to determine the sensitivity of net present value and payback to possible cost overruns.

Moving from this relatively simple situation, a more complex case was written which required numerous decisions to be made by the student while he analyzed case data at the computer terminal. When studying the topic of flexible budgeting the student learned from reading and lectures that budget allowances in businesses should be related to the levels of activity currently being experienced by the firm. Moreover, to accomplish this one must divide accounting costs categories into fixed and variable components, which requires the selection of an activity measure to associate with each cost. This process of choosing an activity measure involved some logical and empirical association of cost variances with the activity levels.

The case required that a flexible budget be prepared for the production department of a manufacturer of beer mugs. All information, including the problem description was stored in the computer system and called by the student. He was given data for four production expenses and four activity measures covering twelve accounting periods. Canned programs for correlation and regression were provided. These were interactive and instructions were provided on their use. A plot program allowed the student to visually scan the relationships between independent variables (activity measures) and dependent variables (expenses).

Solution steps to the case included the selection of an appropriate activity measure for each expense. The student was not only expected to utilize the statistical programs, but to also consider whether the activity measures and expenses having the closest relationship could be logically associated, given the nature of the production process. The following typical problems encountered in practice were built into the data:

1. Pairing the activity measure and expense having the highest correlation resulted in a material negative fixed cost being budgeted.
2. Pairing the activity measure and expense having the highest correlation resulted in an immaterial negative fixed cost being budgeted.
3. An activity measure having a strong logical association with an expense correlated poorly due to a single observation deviating significantly from a rather uniform pattern.

The student was instructed that there was no unique solution to the case, but rather each student was to analyze the data and construct a flexible budget which he felt would be most useful for planning and control. Choices could be made among the activity measures for each expense and one could delete observations from the accounting data (twelve periods provided) in constructing the budget.

Many students, for example, felt that ovenhours was a logical activity measure for coke cost even though the correlation was relatively low as indicated in Exhibit 2. Their judgment was validated when coke cost was plotted as a function of ovenhours (OVHRS) since it was discovered that, if one observation was excluded, a very high correlation existed. To determine which observation of the twelve was the non-representative one, the student could display the data by typing the index of the data (OVHRS and COKE) and then attempt to locate it by inspection. A simpler approach illustrated in Exhibit 2 is to divide coke cost by ovenhours and inspect the unit cost figures, thus the fifth observation is easily distinguishable as the non-representative one.

At this point the student could ask the instructor for an explanation of the unusual cost which occurred in the fifth period. The answer given was that a rare malfunction occurred in the coke burner, thus increasing coke consumption above normal. Students then decided whether to include or to exclude that observation from their budget.

Answers to situations of this kind can vary to illustrate various problems which exist in planning and control. Other possible causes of unusual costs include errors in recording accounting transactions, disruptions in other parts of the production process which impacts on the cost under consideration, irregular outlays which are expected and justifiable, and poor control. The student must decide whether to include such cost observations in his budget, and if so, how to include them.

CAPBUDGET
ENTER THE INVESTMENT OUTLAY
 □: 25000
ENTER THE ANNUAL CASH FLOWS, SPACING BETWEEN AMOUNTS
 □: 2500 4000 6000 8500 8000 7500 4000 1000
ENTER THE INTEREST RATE
 □: .1

 THE NET PRESENT VALUE IS \$2612.08

PAYBACK IS 4.47 YEARS.

CAPBUDGET
ENTER THE INVESTMENT OUTLAY
 □: 500000 550000 600000
ENTER THE ANNUAL CASH FLOWS, SPACING BETWEEN AMOUNTS
 □: 50000 70000 70000 70000 70000 70000 70000 70000 80000 80000 80000 80000
ENTER THE INTEREST RATE
 □: .07

 THE NET PRESENT VALUE IS \$57010.32 7010.32 42989.68

PAYBACK IS 7.42 8.14 8.85 YEARS.

EXHIBIT 1: Illustration of the use of a capital budgeting model.

Note in Exhibit 2 that the mechanical aspects of the analysis were almost effortless in that the student did not have to enter the numerical data, but simply typed the index (name) under which the data was stored. Moreover, the program which displayed the problem also instructed him in the deletion of unwanted data. The second run of the correlation program and the regression program in Exhibit 2 illustrate that the fifth observation is deleted by typing an array, one's for retained and zero's for omitted observations, followed by a slash and the variable name.

The value of the time-shared terminal for such instruction is apparent to one familiar with typical textbook problems on flexible budgeting. Ordinarily, the most complex problem will require two least squares calculations and will not require more than one decision by the student. The case discussed here, in contrast, permits the student to choose among sixteen alternatives, permits him to visually examine (using the plot program) relationships between the data sets almost effortlessly and in real-time, and allows him to use or disregard data at will. Thus he focuses on the concepts rather than being distracted by the computations. This is only possible because of the availability of the time-shared terminal and easy-to-use programs.

The capstone section of the course is a study of corporate financial planning models, their characteristics, users, and limitations. Such models have only recently been developed, but are rapidly spreading in use among the larger and more progressive corporations. At present the student's only exposure to financial planning models is through periodical readings since they have not yet appeared in textbooks. No approximation of such a planning technique was available prior to the development of computerized models. If one looks for an earlier expression of the concept, the hand-generated budget is the best available even though the scope of a financial model far surpasses any such budget both quantitatively and qualitatively.

Following a general study of corporate planning models, the student is given a duplicated description of a company, its objectives, goals, operations, and environment. The packet contains the most recent run of the five-year financial plan, and several proposals concerning new investment outlays and asset redeployments. The financial plan reveals that the corporate objective of sales growth will be met if the plan is adopted and achieved, but that a planning gap exists between planned and target earnings, and between planned and desired corporate liquidity.

A corporate financial planning model is available via the time-shared terminal, which model the student uses to determine the impact of each proposal on the achievement of corporate objectives. Using an interactive program, the student changes only those inputs necessary to effect the changes implied by a proposal and receives as output the resulting five year financial plan - balance sheets, income statements, cash requirements, and analytical ratios. Nor does the student consider the model and the terminal a "black box," since he has previously studied what the model does and how it does it. Thus this final computer exercise and related study afford the student a knowledge of how top management planning is conducted and allows him to use the most powerful planning tool available to corporate management.

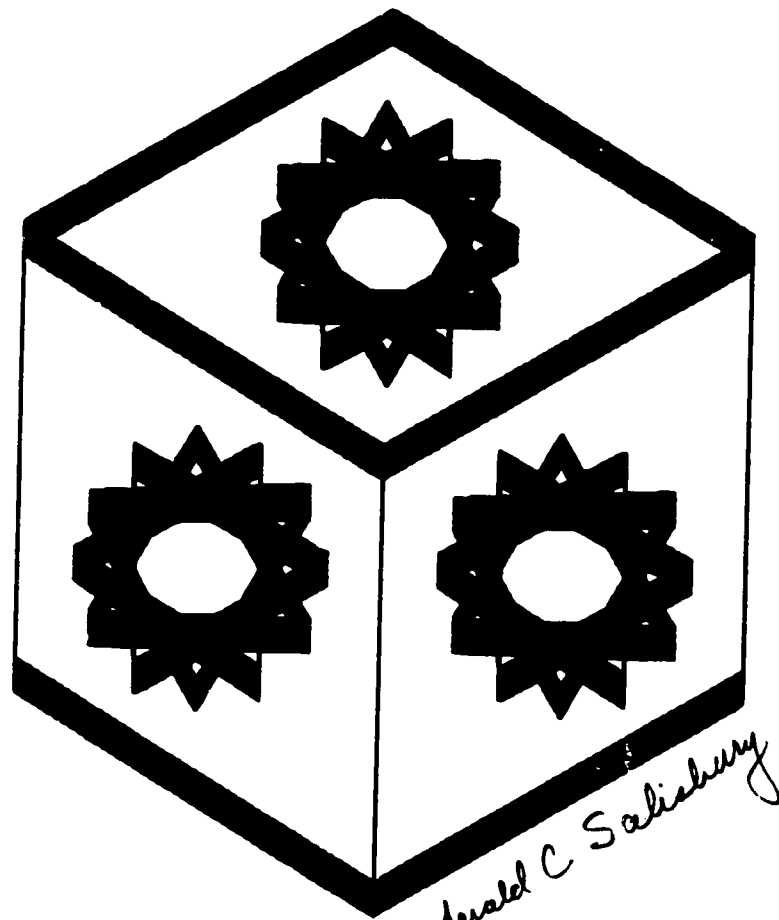
The use of the corporate planning model simultaneously illustrates that businessmen, rather than maximizing some single objective such as profit, have multiple objectives, and that a rather vague, but real trade-off exists between these objectives. The student is not only told through reading and lectures that the management process is heuristic in nature, but through using such solution devices, he experiences a trial and error approach. Devey said that "people believe to the extent that they participate." This is certainly true of students of planning and control systems. Rather than the course being something that the student is told, it is something that he does. And this difference between hearing and doing represents a new dimension in the learning process.

Conclusions

The use of time-shared remote terminals as a student tool in the learning process at The University of Texas at Arlington has resulted in significant improvements in the teaching of corporate planning and control. The student has discovered that statistical tools acquired in earlier courses have practical application in subsequent courses and in addressing practical business problems. Planning and control concepts and models have become an experienced reality rather than something to which he has been "exposed." The complex nature of the management process and ambiguities encountered in business problems are recognized since the student is required to make tentative decisions in solving cases without having all the possible implications stated beforehand. He learns that unique decisions do not exist and are not expected.

This is possible despite the use of the computer which is number-oriented and intolerant of ambiguity, and because the computer frees the scholar to study the discipline free from the fetters of computational limitations.

Thus the realism of the study is a function of the ingenuity of the professor. The challenge is appalling, but the race is stimulating.



Gerald C Salisbury

BOX WITH SPIROGRAPHS by Gerald Salisbury

Problem: Take a spirographic form, develop a new format for representation

THE IMPACT OF COMPUTER-BASED INSTRUCTIONAL METHODS
IN GENERAL CHEMISTRY

J. J. Lagowski, S. J. Castleberry and G. H. Culp
The University of Texas
Austin, Texas 78712
Telephone: (512) 471-3288

Introduction

In the past the use of computers in the educational process has been generally limited by systems and software focused primarily on supporting calculational or data processing efforts. However, recently there have been reports describing the production, use, and evaluation of programs designed for use directly in the instructional process.

There are two major aspects to the educational process; the teacher teaching and the student learning. All too frequently teachers become overly involved in attempting to help students learn in a poor environment, rather than teaching; that is, they have the burden of assigning, grading and giving students feedback on homework and tests; helping students with their assignments; and conducting tutorial/remedial drill group interactions. To a large extent the computer can perform these tasks (on an individual basis) as well as or better than the instructor, for the computer can be programmed to be the world's most patient tutor and has the capability of interacting with the student as often and for as long as the student requires. This, of course, does not diminish the teacher's contact with students but rather, makes it possible for the teacher-student interaction to be richer in the activities which teachers perform best: giving insights into difficult concepts, transmitting an understanding of abstract ideas, inspiring students, and obtaining behavioral objectives in the affective domain.

Computers can be used to individualize student experiences in several ways. Programs can be used to measure entering skills and prescribe a series of programs which contain remedial materials if necessary, standard curriculum materials, and/or advanced placement materials. In addition, each program can branch students ahead, provide extra work or help, or branch back on the basis of the student's aptitude. Well designed programs can allow for individual differences in learning speeds by allowing students to take a module as many times as necessary and work for as long as desired or necessary. When the student's interest dictates, modules which supply specialized or enriched materials can be supplied.

Instructional programs can generally be classified as tutorial-drill, laboratory simulation, or evaluation. With computer-based interactive programs it is possible to provide the student with tutorial-drill materials (practice problem sets, question and answer sessions, problem situations, etc.) which are tailored to his needs and are unique to him. Programs which simulate laboratory experiments can be used to extend a student's laboratory experiences to much greater depth than ever before possible; that is, the student can perform a greater variety of experiments more often if necessary, and each time collect unique data. The time compression/expansion capability of the computer allows the student to perform experiments which in the real world occur on a very short time base or a very long time base (e.g. kinetic studies). In addition, computer simulations allow students to perform experiments which are too dangerous for beginning students to perform on a large scale in the real laboratory and to perform experiments which are too sophisticated and require too expensive an apparatus for wide-scale use by beginning students.

Consider a general chemistry course composed of a lecture segment, a laboratory segment, a reading segment, a homework segment and a testing segment. In lecture, CAI methods can be used to take some of the burden of helping students learn from the instructor [i.e. working examples, problems, illustrating mathematical models, etc., where facilities exist for mass display (overhead projection or closed circuit T.V.)]. In laboratory, computer methods can supply tutorial/drill materials and simulations. In the homework segment, drill and practice and remedial work can be supplied by the computer. In the testing segment, the computer can generate exams, administer and grade minimum level exams and make-up quizzes. In all of the segments, the computer can be used to keep records and calculate grades. Figure 1 summarizes these applications.

If these applications strike you as worthwhile, you may still be wondering if they are economically feasible. This is a question highly dependent upon local conditions and facilities. Where time-sharing computer facilities already exist, these applications are quite feasible and require very little actual computer time. Where there are no prior existing facilities, mini-computers may still provide an economical means of implementing computer methods. Some possible hardware configurations and estimated costs are shown in Figure 2.

Course	Lecture	Laboratory	Readings	Homework	Testing
Supplementation	Examples, Tutorial/Drill, Record keeping, Grading	Tutorial/Drill, Simulations, Record keeping, Grading	Generate Bibliographies	Drill & Practice, Remedial Work Record Keeping Grading	Minimum level make-up, Test generation, Record keeping, Grading

FIGURE 1.

Previous investigations conducted under carefully controlled experimental conditions using general and organic chemistry students have consistently indicated that computer-based instructional techniques exert a positive effect upon student performance in the attainment of course objectives[1-4].

In addition, students have consistently and overwhelmingly indicated positive attitudes and opinions in favor of computer techniques applied to instruction. When computer techniques are presented to students as supplemental material they tend to view them as aids rather than as another dehumanizing barrier between the student and instructor. As a supplement, the computer can remove the burden of stereotyped activities from the teacher and allow him more time for the activities which he performs best.

There are many different ways in which computer methods can be applied to the educational system and many different philosophies motivating and guiding their use. We take the position that computer methods should:

1. Supplement, not replace the existing course.
2. Be designed to help students learn, not necessarily to teach or merely transfer standard information.
3. Help to individualize student experiences.
4. Provide the basis for a self-paced instructional approach for the student.

With these ideas in mind, an experimental Chemistry 302 course was offered in the Fall of 1971, with the following characteristics:

Course Description

The existence, or creation of instructional material for use with computers automatically leads to a modularization of the subject. Accordingly the experimental 302 course appeared from the student's view in modular form. The conventional three contact-hours per week were designed as one hour per week in formal lecture, small group discussion, and at a computer terminal. The first two interactions were at the time scheduled for the course; one hour of terminal time was assigned at the convenience of the student. Students could have more computer time on a first come - first serve, sign-up basis.

The course content was that agreed upon with the coordinator, viz., Chapters 7, 9, 10, 11, 13, 14, and 15 in Slabaugh and Parsons.

Lectures. One hour per week was used in formal lecture to present broad concepts and attempt to draw apparently diverse material together, but with a minimum of examples being worked in class. A ten-minute quiz was given at the start of each lecture over the material of the previous week. Occasional demonstrations were performed.

Discussion Periods. This hour was devoted to student discussion and problems. The students dictated the course of the discussion, and, if desired, problems were worked out in detail. No attempt was made to lecture during this hour.

Student Terminal Time. The assigned hour at a terminal[5] was used by students working with programs classified as (a) tutorial, (b) simulated experiments, or (c) examination. A list of the programs in categories (a) and (b) appears in Table 1. The programs in category (c) were essentially drill and practice problems similar to those given on the 10 minute quizzes and

TUTORIAL MODULES

<u>Chapter</u>	<u>Code</u>	<u>Description</u>	<u>Points</u>
7	CHEM1	The Gas Laws	15
9	CHEM114	Henry's and Raoult's Law	10
10	CHEM116	Colligative Properties	10
10	CHEM113	Solution Concentration	10
10	CHEM2	Solution Stoichiometry	10
11	CHEM119	Equilibrium	15
13	CHEM107	pH, [H ⁺], pOH [OH ⁻]	10
13	CHEM124	Common Ion Effect	10
13	CHEM126	K _{sp}	10
14	CHEM36	Redox Equations	10
15	CHEM109	Elementary Thermochemistry	10
15	CHEM139	Thermochemistry	15

SIMULATED EXPERIMENTS

7	CHEM3	Molar Volume of N ₂	25
10	CHEM115	Colligative Properties	20
11	CHEM32	Reaction Kinetics	25
13	CHEM122	pH and K _a Determination	25
13	CHEM19	Titration	20
14	CHEM127	Faraday's Law	20
15	CHEM41	Calorimetry	25

TABLE I

major examinations. Copies of typical student interactions for these programs are available for inspection.

In this experiment, there was no limit placed upon the number of times a student might interact with any of the programs described. That is, he could work at the material until he reached the level of achievement he desired.

This course, in addition to presenting the student with self-paced modules, was organized about the general principles of contingency management[6]. Briefly, a relationship is established between a response and a subsequent event. Responses correspond to student behavior and the subsequent events are the reinforcers for this behavior. Classically these are the grades given at the end of a semester after the student has performed to some level of accomplishment. As has been stated: "Final grades, like most social rewards, however, are large events that are awarded after a long delay and not in any specific relation to easily identifiable responses. In order to be useful with a contingency management system, the final grade must be divided into smaller parts which can be made available to students throughout the semester." [6] One obvious way of accomplishing this end in a modularized course is in terms of the total accumulated points from each module completed to a given level of achievement. Accordingly, each module was assigned a point value which reflected its relative contribution to the total course material (Table 1). In addition points were assigned for (a) attendance (which was not required) at lecture and discussion, (b) quizzes, (c) major examinations. The point schedule is given in Table 2. Grades were assigned on the following basis: A = 100-90%, B = 89-80%, C = 79-70%, D = 69-60%, F = below 60%.

Three major examinations (two hours each) were given at night. Each of these examinations covered about 1/3 of the material in the course. There was no comprehensive final examination.

Results

Two traditional 302 courses were used for comparison with the computer-supplemented 302 course. Figure 3 and 4 show the Standard Achievement Math Test and Standard Achievement Verbal Test score distribution for the three sections. From these we see that the three sections have approximately the same range and distribution of scores.

In Figures 5 through 7, score distributions of the quizzes and tests are shown, a distribution for the paper-pencil tests and a distribution after students have had an opportunity at computer make-up. Figure 8 summarizes the means on the tests and quizzes.

Figure 9 gives an indication of the degree of use of the computer make-up as well as the kinds of students using the make-up. Figure 10 indicates the usage of the computer tutorial and simulation modules.

Figure 11 shows the grade distribution in the computer-supplemented section and the two comparison sections. In Figure 12, a comparison of the drop rates between the supplemented section and one traditional course is shown.

In Figures 13 and 14, cost factors are tabulated. Figure 13 shows authoring and debugging costs while Figure 14 shows student usage costs.

Discussion

In this experiment it is evident that the use of supplemental computer-based instructional techniques exert a positive influence upon student performance. While the SAT scores for the experimental group result is essentially a normal distribution, as seen in Figures 3 and 4, it is interesting to note in Figures 5 and 6 that the distribution of scores for TEST 1 and TEST 2 prior to the computer make-up indicate a general shift toward a higher achievement level. TEST 1 and TEST 2, in comparison with TEST 3, covered areas that had a higher degree of computer-supplemented materials available for student use. The results of the latter examination (Figure 7) appear to reflect this. Thus a comprehensive comparison of the three exams tends to indicate a degree of correlation between achievement and access to computer-supplemented instructional techniques.

The influence of self-paced individualized computer-based examination make-up is dramatically seen in Figures 5 through 8. In every instance of student performance evaluation, the existence of computer-based instructional techniques allowed a much greater number of students to attain a high degree of proficiency in specified behavioral objectives. The majority of students making use of the computer examination make-up appear to be either the lower or slower achievers within the class (see Figure 9), further illustrating the marked effect of computer-based self-paced individualized instruction on the instructional process. The carry-over of these results is clearly evident in Figure 11. Seventy percent of the experimental class gained at least a 90% proficiency in the course objectives, while the traditional classes indicate a more normal distribution of semester grades.

GRADING SCHEDULE

ITEM	MAXIMUM POINTS
3 MAJOR EXAMINATIONS (100 each)	300
10 QUIZZES (10 each)	100
12 DISCUSSION PERIODS (3 each)	36
15 LECTURES (3 each)	45
7 SIMULATED EXPERIMENTS	160
12 TUTORIAL MODULES	135
TOTAL	776

100 - 90% = A
 89 - 80% = B
 79 - 70% = C
 69 - 60% = D
 Below 60% = F

TABLE 2

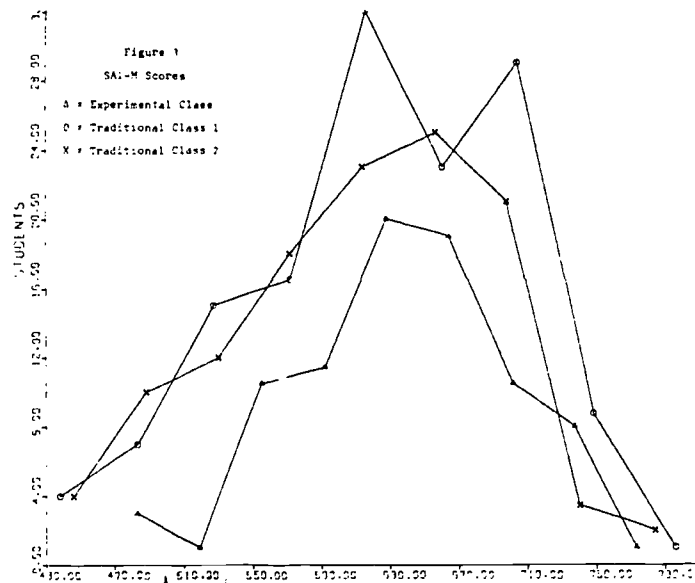


FIGURE 3

TEST AND QUIZ MEANS

	T1	T2	T3	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
TEST MEAN	60.4	62.9	43.3	5.7	5.4	5.3	5.2	6.7	5.7	4.4	5.5	6.5
TEST MEAN AFTER COMPUTER MAKE-UP	85.1	90.1	81.1	9.0	8.4	8.4	9.3	12.8	9.8	8.7	8.4	8.9

FIGURE 8

COMPUTER TEST USAGE

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	T1	T2	T3
# STUDENTS TAKING TEST	113	115	111	109	105	102	100	84	99	115	110	100
# STUDENTS TAKING COMPREHENSIVE MAKE-UP	50	59	67	52	47	39	52	35	35	36	73	
# STUDENTS SCORING BELOW 70%	57	82	70	54	103	45	89	57	42	72	60	
# STUDENTS BELOW 70% TAKING COMPREHENSIVE MAKE-UP	36	44	48	41	47	27	52	27	31	30	51	

FIGURE 9

COMPUTER MODULE USAGE

MODULE	# STUDENT INTERACTIONS
CHEM1	105
CHEM114	103
CHEM116	101
CHEM113	103
CHEM2	98
CHEM119	94
CHEM107	98
CHEM124	87
CHEM126	91
CHEM36	84
CHEM109	69
CHEM131	46
CHEM3	101
CHEM115	95
CHEM32	93
CHEM122	87
CHEM19	88
CHEM127	87
CHEM41	1

FIGURE 10

COURSE GRADE DISTRIBUTIONS

	A	B	C	D	F
COMPUTER SUPPLEMENTED*	70%	9.7%	6.1%	1.2%	13%
TRADITIONAL CLASS *	10.4%	25.6%	18.4%	14.4%	31.2%
TRADITIONAL CLASS	18.7%	33%	26.4%	12.7%	9.2%

* SAME INSTRUCTOR

FIGURE 11

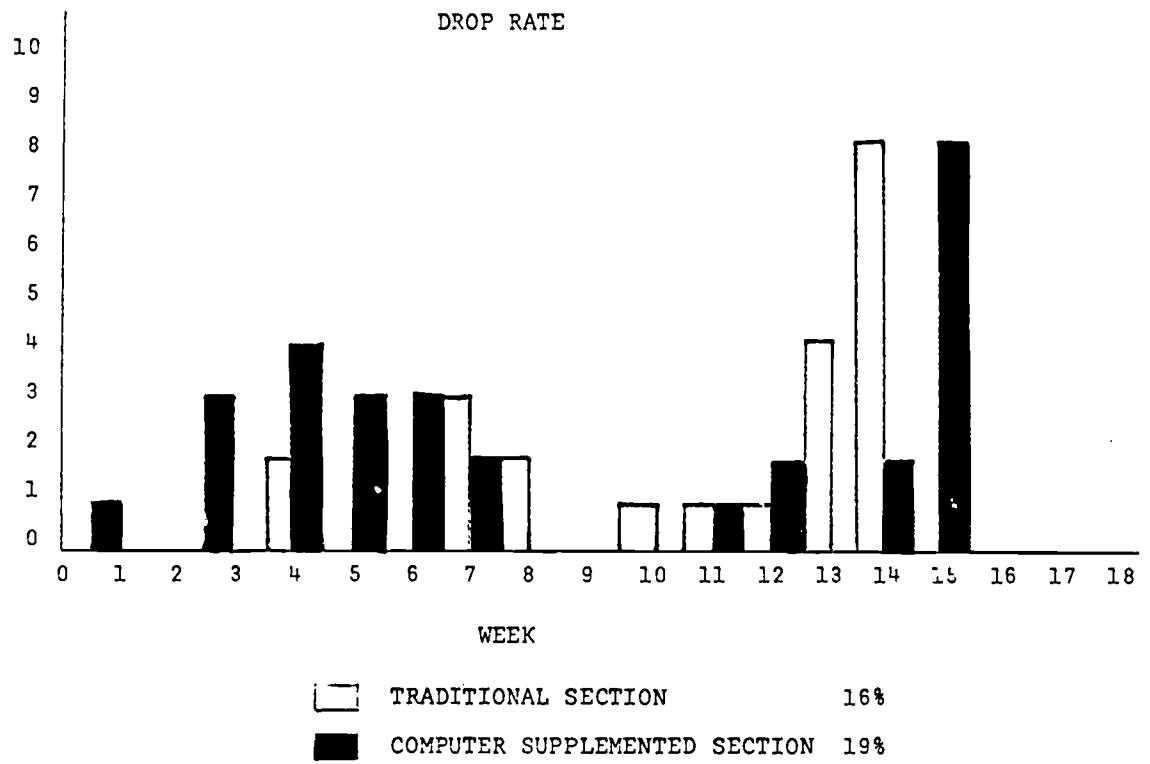


FIGURE 12

97

AUTHORING & DEBUGGING COSTS

# MODULES	TM. HRS.	COMPUTER COSTS	SUPPLIES	PERSONNEL ^b COSTS
86 ^a	11.097	\$2885.36	\$596.87	\$14,000

- a. Equivalent to 44 student-terminal hours of instruction.
- b. Twelve man-months of authoring and content debugging; 9.5 man-months of coding, debugging, and general management.

FIGURE 13

STUDENT USE COSTS

#JOBS	TM HRS*	COMPUTER COSTS	LINE COSTS	STUDENT HRS
2291	18.944	\$4925.48	1613.57	1613.57
			RATIO	
		COST/STUDENT HR	STUDENT HR/TM HR	
		4.05	85/1	

* TM HRS = CPU HOURS + PERIPHERAL OPERATIONS TIME FACTOR
 COMPUTER COST FIGURED ON THE BASIS OF \$260/TM HR

FIGURE 14

It may be seen in Figure 12 that the drop rates of the traditional and experimental courses are essentially the same, except for a slight increase in drop rate for the experimental class following TEST 1.

Figure 13 summarizes the developmental costs for 86 computer modules. These figures indicate a cost of approximately \$200/module, including computer line time, supplies and salaries for personnel. It is important to note that very careful planning and design go into the development of a module, and its lifetime should be considerable. Thus, following initial development of a module, the only additional cost comes from up-dating or revision and would be negligible.

In Figure 14, the cost of 2,291 student interactions is summarized. The figures indicate a cost of approximately \$4.00 per hour of student connect time, or about \$2.85 for each student interaction, based upon a line charge of \$1.00 per connect hour and rate of \$260.00 per computed TM hour.

Summary

The development of computer-based instructional techniques to augment a traditional course led to modularization of the course. This, in turn, resulted in a clear and comprehensive description of the course in terms of specific behavioral objectives. The combination of modularization and computer-based instructional techniques provides a unique approach to self-paced, individualized instruction.

The results of the study described in this paper indicate that this combination of course design and instructional techniques yields a very positive effect upon student performance. This effect is particularly evident in the use of computer-based examination make-up for the attainment of specified behavioral objectives. Seventy percent of the class attained a 90% or better proficiency in the course objectives.

The cost of development averaged approximately \$200.00 per module and the cost of student utilization averaged about \$4.00 per connect hour. A total of 86 modules have been developed for use in undergraduate chemistry instruction.

ACKNOWLEDGEMENT

This study was conducted under grants from the Moody Foundation and the Esso Education Foundation. Special acknowledgement and gratitude is hereby extended for their generosity and belief in educational evolution.

REFERENCES

1. S. J. Castleberry and J. J. Lagowski, "Individualized Instruction Using Computer Techniques," J. Chem. Ed. **47**, 91 (1970).
2. S. J. Castleberry, E. J. Montague, and J. J. Lagowski, "Computer-Based Teaching Techniques in General Chemistry," J. Res. Science Teach. **7**, 197 (1970).
3. L. H. Rodewald, G. H. Culp, and J. J. Lagowski, "The Use of Computers in Organic Chemistry Instruction," J. Chem. Ed. **47**, 134 (1970).
4. G. H. Culp and S. J. Castleberry, "Computer-Assisted Instruction in Undergraduate Organic Chemistry: An Evaluation of Selected Programs," Science Education, **423** (1971).
5. Two types of terminal devices were used, teletypes and a teletype-slide projector combination. In the latter case it was possible to include visual displays (e.g., colors of indicators at different pH's) in the programs. The slide projector was computer-controlled.
6. K. E. Lloyd, "Contingency Management in University Courses," Educational Technology, April, 1971.

COMPUTER-AIDED CLASSROOM CHEMISTRY INSTRUCTION VIA
INSTANTANEOUS VIDEO PROJECTION OF TELETYPE OUTPUT

Ronald W. Collins
Eastern Michigan University
Ypsilanti, Michigan 48197
Telephone: (313) 487-0106; 487-0423

The use of computers for instructional purposes in the chemistry curriculum normally involves a 1:1 student-computer interaction. This communication can be either via batch processing or remote teletype in time-shared mode. Furthermore, the role of the machine can range from that of surrogate teacher to simply a high-speed calculator performing data reduction; however, all forms of educational computer utilization can be broadly classified as computer-aided instruction (CAIDI). This general classification can be further subdivided into computer-assisted instruction (CAI), computer-evaluated instruction (CEVIN), and non-interactive computer application (NICA). The criteria used for determining these categories as well as their relationships one to another are presented in Table I. Further information on this classification scheme for computer usage is given in a previous review by this author[1]. In particular, this earlier review covers methods for teaching programming languages and optimizing the role of NICA in the chemistry curriculum; consequently, these topics will not be discussed in the present paper. Instead, the emphasis will be on techniques and pedagogical strategies for "group" CAIDI; i.e., the effective use of on-line computing in the classroom as a component of the normal lecture environment. This usage is neither clearly CAI nor NICA, but rather incorporates some of the features of each. The extent to which on-line classroom computing becomes at least "pseudo CAI" depends on the extent to which the program used includes interrogational features and remedial branching. Without these characteristics the principle goal is the expansion or clarification of course content and hence the category is more properly NICA. In either case the machine is now simultaneously interacting with an entire class and hence the designation "group CAIDI."

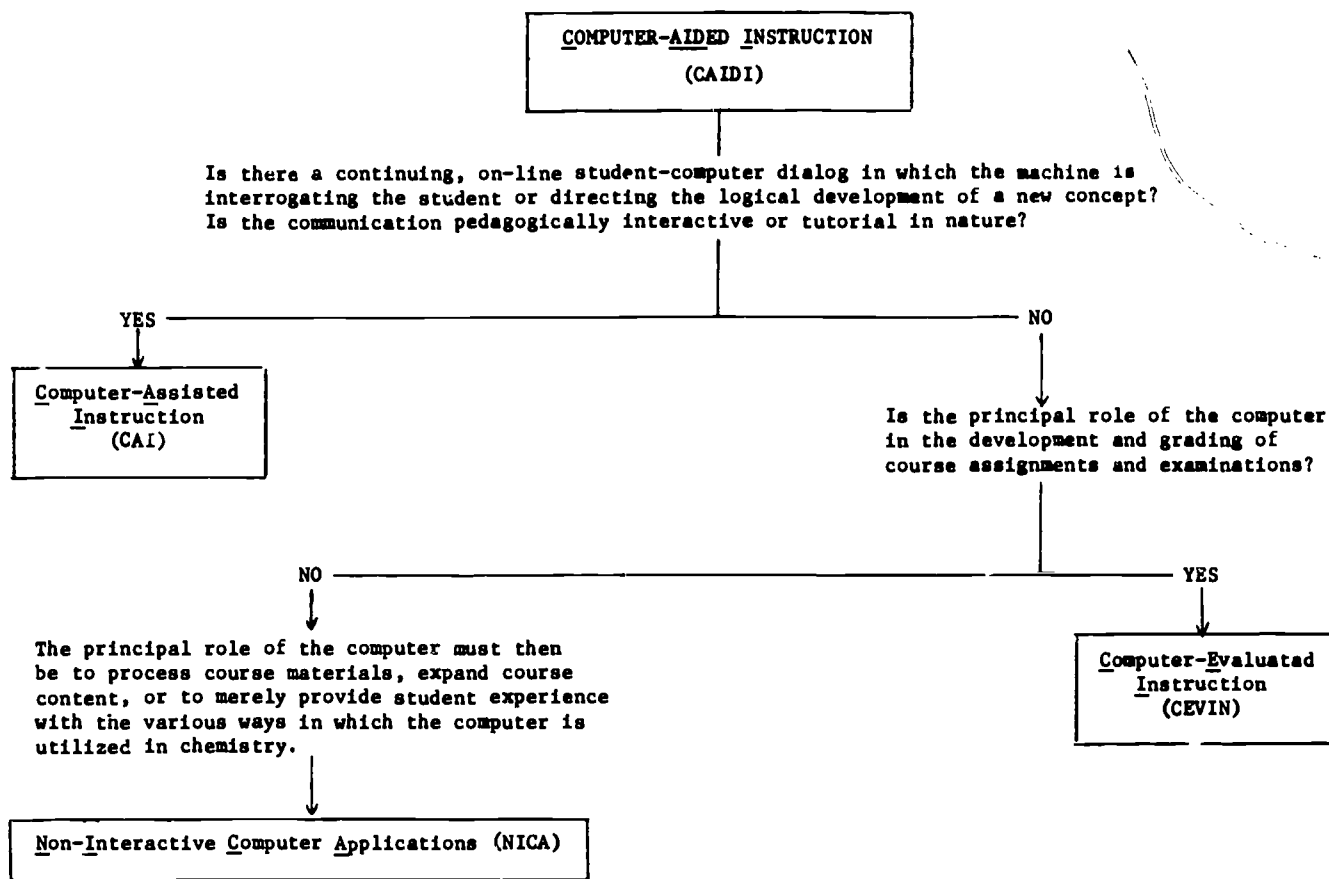
To date, on-line computing during a lecture class has not been extensively used, primarily because of technological problems associated with presenting the results in a visual form suitable for large audiences. However, the recent development of a low-cost adapter[2] for instantaneous video projection of teletype output has made on-line computing a more practical classroom technique. By having a room equipped with only a telephone line and suitably-sized television sets an instructor can now routinely supplement his lectures with the results of on-line computing provided a portable teletype terminal is brought into the classroom. Actually, this configuration for on-line lecture hall computing is only one of several possible arrangements. A summary of the various possible configurations is given in Table II. It would seem, however, that this method employing the video adapter is definitely superior to, as well as being much more convenient than, the use of either a television camera or overhead projector as the transmitting link into the viewing medium. Utilizing a TV camera effectively requires that the instructor have a technical assistant present during his lecture and essentially converts the classroom into a television studio which distracts somewhat from the desired educational environment. Having the teletype output printed directly on plastic transparency material which is then projected onto an opaque screen via the use of a standard overhead projector is technically feasible and commercially available in the form of a teleprinter projector[3], but the clarity of the output for mass viewing does not compare favorably with the video technique.

The video adapter currently under discussion is compact, measuring 3 x 6 x 12 inches, and is equipped with the appropriate cables for easy input connection to a model 33 teletype and output connection into a video receiver. This commercially available device is low-cost and comes equipped with a switch which permits the user to disable the teletype and merely get the video output without accompanying hard copy. The principal advantage of this capability is that the teletype is silent and the lecturer need not speak over the background of mechanical noise. The disadvantage, of course, is that there is no permanent record of the results. The adapter comes in several models, but the one currently being used for chemistry instruction at Eastern Michigan University is capable of displaying an image consisting of 8 vertical lines of output, each 32 characters in length. This has provided excellent viewing in a classroom with approximately 100 seats and 4 ceiling-mounted 25-inch television sets.

Turning now from the technical aspects of on-line classroom computing to the pedagogical aspects, it might be well to begin by summarizing the advantages of this instructional technique:

1. Provides for rapid, accurate calculations and neat, orderly display of the results.
2. Permits the use of sophisticated, accurate

TABLE I
CLASSIFICATIONS FOR INSTRUCTIONAL COMPUTER USAGE



mathematical methods rather than the approximations often used for convenience.

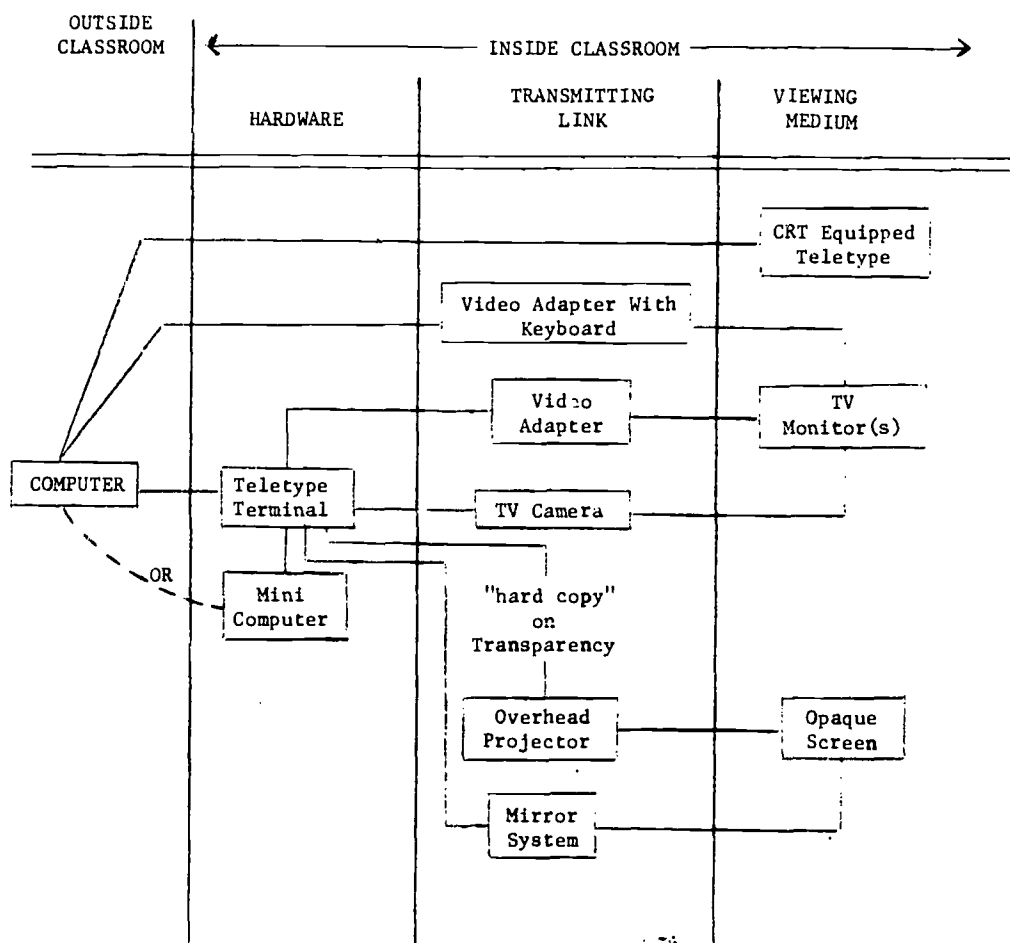
3. Introduces the students to computer programs which can subsequently be used for outside assignments.
4. Provides a focus for active classroom discussion; i.e., via a multimedia three way instructor-student-computer interaction.
5. Brings a dynamic, modern "instrument" into the classroom.

Based on our experience to date at Eastern Michigan University the classroom computing technique is very effective provided that:

1. It is not used too frequently: i.e., if on-line computing is used during every lecture in a given course, then just as with any other instructional aid, students will become somewhat bored with its use.
2. The applications are carefully selected so as to truly reinforce or expand upon some topic under discussion. It is imperative that the data reduction or simulations performed be those which truly require mathematical accuracy not otherwise obtainable by approximation methods, or require the rapid use of mathematical operations (e.g., iteration techniques) not otherwise possible in real time during a lecture. In other words, the computing medium is not the complete message in itself, and the technique must not be exploited purely for its theatrical value during a lecture.

TABLE II

POSSIBLE CONFIGURATIONS FOR ON-LINE CLASSROOM COMPUTING



3. A rapid response time-sharing network is employed which provides access to a reasonably large library of appropriate stored computer programs.
4. The students have access to the same programs for use outside of the classroom. This can be accomplished either by using the identical program in time-shared mode or by providing an analogous batch version of the program. In either event it is very important that any interested student be able to experiment on his own with the same type of calculation performed during the lecture. This not only permits repetition if necessary but it also allows the student to pursue other examples of his own choosing.

The potential applications of this technique for the classroom use of on-line computing in chemistry education are probably unlimited and certainly too numerous to list; however, a few examples are given in Table III. For more exacting delineation of the types of programs involved, two examples which have been used quite successfully in our general chemistry course are shown in Figures 1 and 2. Each figure contains both a listing of the program and a sample output. The first example is on the topic of ionic equilibrium and the program has been written to provide a comparison of the pH values calculated for an aqueous solution of a weak acid, depending on whether a simplifying mathematical assumption is employed or the complete quadratic equation is solved exactly. This permits a rapid, accurate demonstration of the influence which the magnitudes of both the ionization constant and the concentration of the acid have on the validity of the assumptions often used to avoid solving a quadratic equation in this type of equilibrium problem.

TABLE III
SAMPLE UTILIZATIONS OF ON-LINE CLASSROOM COMPUTING
IN THE CHEMISTRY CURRICULUM

<u>Topic</u>	<u>Mathematical Operation (s) and/or Computing Mode</u>	<u>Course(s)</u>
1. Ionic equilibrium calculation	Data reduction via exact solution of quadratic equation	general chemistry quantitative analysis
2. Real gases	Data reduction via iteration solution of higher order polynomials	general chemistry physical chemistry
3. Thermodynamics	Rapid data reduction using stored database	general chemistry physical chemistry
4. Titration phenomena	Curve fitting, analysis of data via derivative approximations, graphical simulation	general chemistry analytical chemistry
5. Identification of unknowns	Data evaluation via comparison to stored database	organic qualitative analysis x-ray powder diffraction
6. Information retrieval	Keyword searching of stored database	chemical literature

The second sample program is concerned with the topic of real gases as depicted by the van der Waal's equation in comparison to the more-frequently used ideal gas model. Again, the program is constructed so as to permit a rapid, yet accurate comparison of the answers obtained from the two possible equations; i.e., the real gas equation and the $PV=nRT$ ideal gas version. The mathematical approach used in the program is to first solve the ideal gas equation for the apparent volume occupied by a given quantity of gas under specified conditions of pressure and temperature, and then to use this ideal value as the first approximation for an iterative solution of the more exacting van der Waal's equation. In this way it is easy to dramatize to the students those physical conditions which either maximize or minimize real gas deviations from ideality. Furthermore, when used in a freshman level general chemistry course, this program introduces the students to the powerful numerical method of iteration. Although the students may not fully understand the mathematics they will certainly appreciate the obvious value of the iteration technique for chemistry applications.

FIGURE 1

IONIC EQUILIBRIUM PROGRAM

```
10' THIS PROGRAM WAS DEVELOPED FOR ON-LINE CLASSROOM COMPUTING
15' BY SUDHIR VAIDYA UPON DR. COLLINS' REQUEST.
25 REAL QH30,QPCT,QPH
29 PRINT " FOR HOW MANY ACID CONCENTRATIONS WILL PH BE CALCULATED?"
30 INPUT,N
40 AK=1.8E-5
49 PRINT " WHAT IS YOUR INITIAL ACETIC ACID CONCENTRATION IN M/L?"
50 INPUT,HAC
55 PRINT " WHAT IS YOUR CONCENTRATION INCREMENT?"
56 INPUT, X
60 B=AK*AK
61 PRINT
62 PRINT 105,
63 105 FORMAT (9X," PH"6X,"EXACT PH"/)
70 DO 101 I=1,N
80 TEMPI=AK*HAC
90 H30=SQRT(TEMPI)
100 PCTION=100.0*(H30/HAC)
120 PH=(LOG(H30))/2.3026*-1.0
130 TEMP2=B+(4.0*HAC*AK)
140 QH30=(-AK+SQRT(TEMP2))/2.0
150 QPCT=(QH30/HAC)*100.0
160 QPH=LOG(QH30)/2.3026*-1.0
170 PRINT 102, I,PH,QPH
180 102 FORMAT (1X,13,4X,F7.4,4X,F7.4)
190 HAC=HAC-X
200 101 CONTINUE
210 END
```

PHHAC 12:49 01/21/72 FRI.

```
FOR HOW MANY ACID CONCENTRATIONS WILL PH BE CALCULATED?
? 5
WHAT IS YOUR INITIAL ACETIC ACID CONCENTRATION IN M/L?
? 0.001
WHAT IS YOUR CONCENTRATION INCREMENT?
? 0.0001
```

	PH	EXACT PH
1	3.8723	3.9015
2	3.8952	3.9259
3	3.9208	3.9533
4	3.9498	3.9846
5	3.9833	4.0208

104

FIGURE 2

GAS LAW PROGRAM

```
5' VAN DER WAALS EQUATION
10' THIS PROGRAM WAS DEVELOPED BY SUDHIR VAIDYA UPON
15' DR. COLLINS' REQUEST IN JULY 1971.
16 DIMENSION GAS(1)
20 PRINT
30 PRINT " SOLUTION OF VAN DER WAALS EQUATION BY ITERATION"
31 PRINT
32 PRINT
80 PRINT " TYPE THE NAME OF THE GAS."
82 INPUT 2002, GAS
83 2002 FORMAT (A6)
85 PRINT 2001, GAS
87 2001 FORMAT(IX,"VAN DER WAALS CONSTANTS A AND B FOR"2X,A6,"ARE")
89 INPUT, A,B
90 PRINT " TYPE THE VALUES OF - T,P,N,CONVERGENCE FACTOR AND THE
91 + MAXIMUM NUMBER OF ITERATIONS."
100 INPUT,T,P,SN,C,MNI
110 N=1
115 R=0.082056
120 V=SN*R*T/P
130 PRINT 1001,V
140 1001 FORMAT (/IX," USING IDEAL GAS EQUATION"/IOX,"V= "F10.6//)
150 1808 VW=(SN*R*T)/(P+A*SN*SN/(V*V))+SN*B
160 PRINT 1002,N,VW
170 1002 FORMAT (I4,5X,F10.6)
180 D=ABS(VW-V)
190 V=VW
200 N=N+1
210 IF (D-C)999,999,1003
220 1003 IF (N-MNI) 1808,1808,1010
230 1010 PRINT " NO CONVERGENCE."
240 999 END
```

SOLUTION OF VAN DER WAALS EQUATION BY ITERATION

TYPE THE NAME OF THE GAS.

? CO2

VAN DER WAALS CONSTANTS A AND B FOR CO2 ARE

? 3.59,0.0427

TYPE THE VALUES OF - T,P,N,CONVERGENCE FACTOR AND THE MAXIMUM NUMBER OF ITERATIONS.

? 50,10,10,0.000001,100

USING IDEAL GAS EQUATION

V= 4.102800

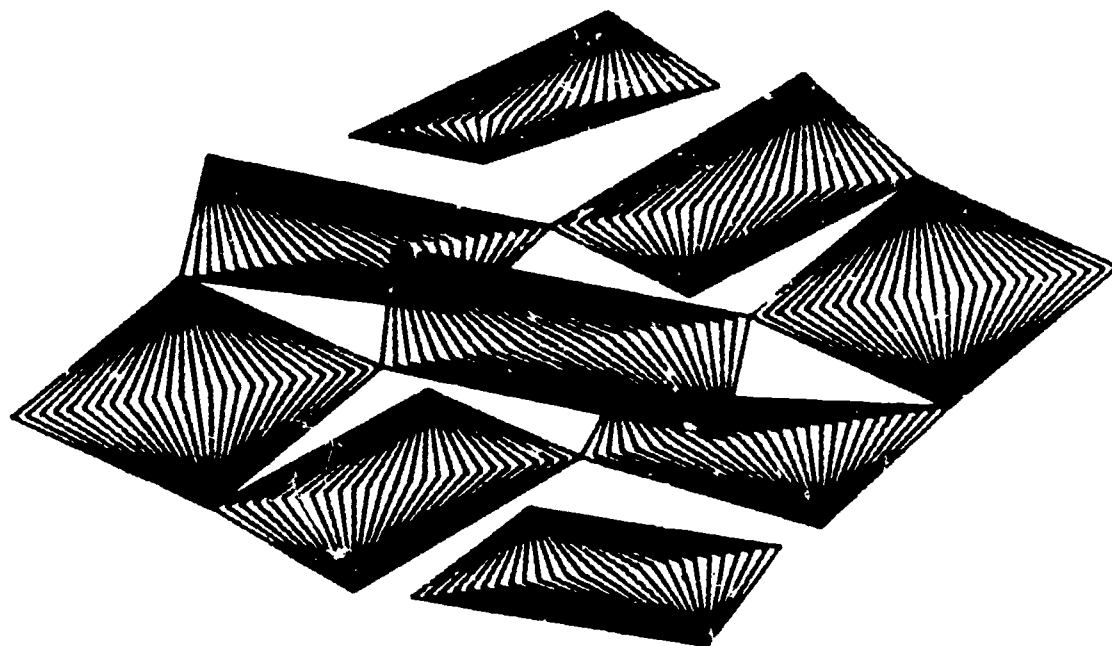
1	1.736661
2	.744967
3	.489459
4	.454198
5	.450442
6	.450058
7	.450019
8	.450015
9	.450014

In summary, on-line classroom computing is a valuable asset to teaching and a modern technological development which should not be overlooked, particularly in any institution where time-shared computing is available and where television facilities are either permanent classroom fixtures or are available in portable models. In such an environment the only required installation is a telephone line into the lecture room. The cost of the described adapter is relatively modest and the physical problems associated with moving and handling the components of the apparatus; i.e., teletype, adapter, and the video sets (if not permanently mounted) are not insurmountable. In fact, with a little practice, the equipment problems are no greater than those usually encountered with the use of movie projectors, and the dynamic, real-time, unrehearsed nature of the computing approach is pedagogically much more rewarding. The role of the television display of the computed output as a focus for classroom discussion also cannot be overemphasized. Asking for student responses regarding appropriate input parameters and/or output values often promotes extremely active, and sometimes controversial, discussions in the classroom. Finally, this method displays the power of computing as a modern, relevant scientific and instructional tool to mass audiences, some of whom may not otherwise be exposed at all to computers and computing technology.

REFERENCES

1. R. W. Collins, "Teaching Programming Languages and Optimizing Non-Interactive Computer Applications (NICA) in the Chemistry Curriculum," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, 99-110 (1977).
2. Manufactured by Ann Arbor Terminals, Inc., 6107 Jackson Road, Ann Arbor, Michigan.
3. Manufactured by I. P. Sharp Associates Limited, Computer Products Division, 320 Queen Street, Suite 2206, Ottawa 4, Ontario CANADA.

100



RANDOM FORMS by Michael Seaters
Problem: Imaginative flight form development

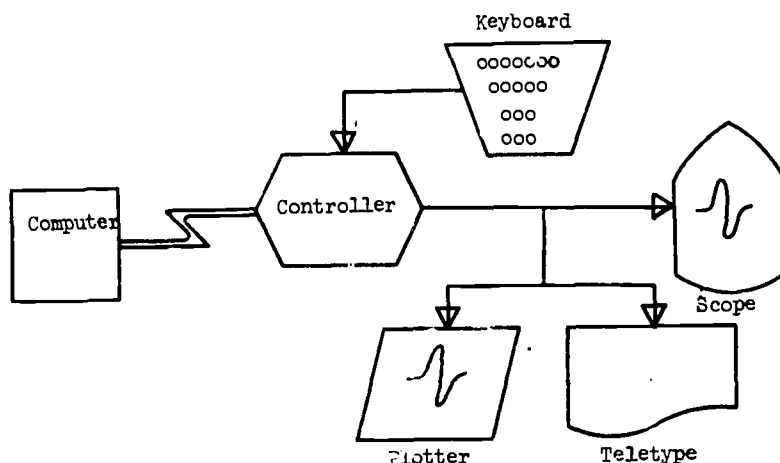
PITT'S INTERACTIVE GRAPHICS AND
COMPUTER-GENERATED REPEATABLE EXAMINATION SYSTEMS

W. F. Sliwinski and K. J. Johnson
University of Pittsburgh
Pittsburgh, Pennsylvania 15213

Several applications of computers in chemical education have been tried at Pitt during the past 3 1/2 years. These include CAI, use of canned programs for data reduction and simulation, use of an interactive graphics system, and teaching computer programming as part of the undergraduate curriculum[1]. There is evidence that the impact of these efforts on the quality of our undergraduate instruction is favorable, so these projects are being continued and extended. This paper describes two new projects: A teletype-oriented interactive graphics system and a question-format based computer-generated repeatable examination system.

Interactive Graphics

One of the earliest interactive graphics systems is the Culler-Fried System at the University of California at Santa Barbara[2]. The NSF recently supported a network consisting of ten universities around the country connected via phone lines to the UCSB Computer Center. The graphics terminal configuration is shown below.



The grant provided the hardware, communication costs, computer time, etc. for one year. We are now developing the software and hardware required to provide essentially the same graphics package on Pitt's PDP-10 time-sharing system. The software problem was tackled first.

We have written an interactive graphics package, modelled after the Culler-Fried system, in FORTRAN. The program reads a 72 character line of instructions from a teletype terminal, decodes the line, executes the instructions, and requests the next line. The following example illustrates the power of the system for simulation applications.

Two-site NMR exchange, given several simplifying conditions, gives the following line shape function, $I(\nu)$:

$$I(\nu) = \frac{K (\nu_A - \nu)^2}{\left[\frac{1}{2} (\nu_A + \nu_B) - \nu \right]^2 + 4\pi^2 \tau^2 (\nu_A - \nu)^2 (\nu_B - \nu)^2}$$

where K is a normalizing constant (taken as unity), τ is the lifetime, ν_A and ν_B are resonance frequencies of sites A and B, and ν is the frequency[3]. If $2\pi\tau(\nu_A - \nu_B) > \sqrt{2}$, two lines result. The two peaks coalesce when $2\pi\tau(\nu_A - \nu_B) = \sqrt{2}$, and sharpen as this value decreases. In this example $0 \leq \nu \leq 100$, $\nu_A = 33.3$, $\nu_B = 66.7$ and τ is an input variable. The following routine is a user program stored on the disk.

- | | |
|-------------------------------|------------------------|
| A) 1. TITANIUM(IV) CHROMATE | 2. URANIUM(VI) SULFITE |
| 3. NICKEL(II) CARBONATE | 4. GOLD(III) NITRIDE |
| B) 1. TITANIUM(IV) DICHROMATE | 2. URANIUM(VI) SULFITE |
| 3. NICKEL(II) CARBONATE | 4. GOLD(III) NITRIDE |
| C) 1. TITANIUM(II) CHROMATE | 2. URANIUM(VI) SULFATE |
| 3. NICKEL(II) CARBONATE | 4. SILVER(III) NITRIDE |
| D) 1. TITANIUM(IV) CHROMATE | 2. URANIUM(VI) SULFITE |
| 3. NICKEL(II) CARBONATE | 4. SILVER(III) NITRIDE |
- E) NONE OF THE ABOVE.

The FORTRAN program consists of a BLOCK DATA subprogram, the main program and one subroutine for each item. The BLOCK DATA subprogram initializes several arrays that are in COMMON. These include element names, symbols, atomic weight, cation and anion names, etc. The main programs contain random number initialization instruction, reads a key for generation of the test, and calls the desired subroutines.

The primary objective of this project is to allow students in the first term of general chemistry to take the first hour exam on a repeatable basis. The computer can generate hundreds of examinations of equivalent difficulty. These can be administered by proctors in an examination room that is available several hours a day. The proctor can perform an item analysis as he grades each test using the computer-generated answer key. The item bank can thus be proved and revised as desired.

The system has a number of other applications. Students may use it in a tutorial mode to drill selected topics. Items may be generated for recitation quizzes, parts of hour exams and final exams. The system can generate tests for screening and diagnostics purposes. It can be used to generate a make-up exam of arbitrary difficulty. It can be used to help students identify this weakness in assumed background material, etc.

Conclusion

The computer-oriented curriculum development project at Pitt now has three areas of emphasis: tutorials, simulations and testing. The graphics package described here provides a very powerful language with graphics output for simulation and data reduction applications. The CGRE system is adding a new dimension to our CAI tutorial-drill package and will allow us to restructure the general chemistry curriculum.

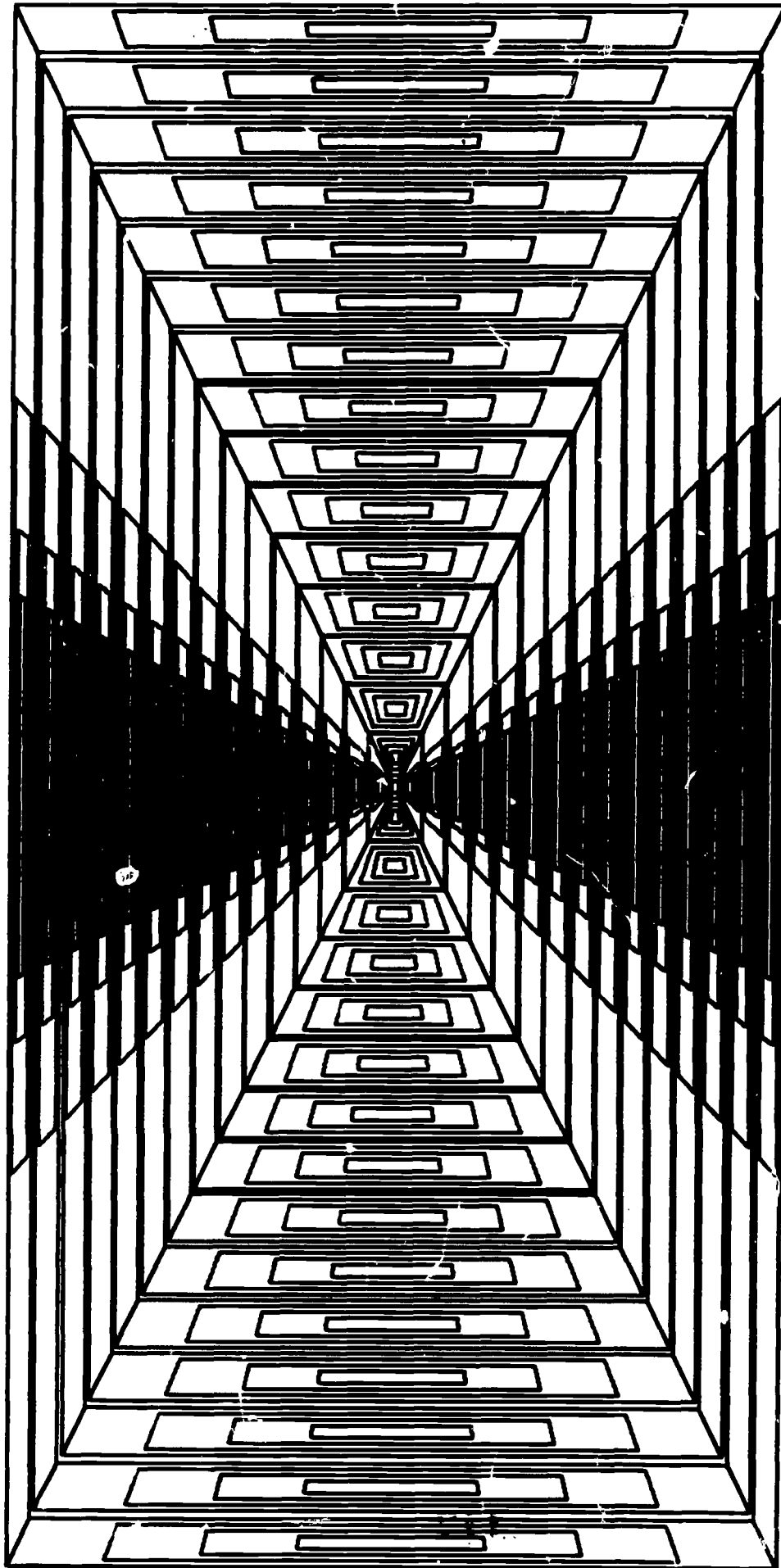
ACKNOWLEDGEMENTS

Support from the National Science Foundation Grants GJ-696 (graphics network) and GU-3184 (institutional science development) is gratefully acknowledged. The latter grant provided postdoctoral support for WFS. Prof. L. E. Epstein has contributed a number of questions to the CGRE system. S. Levitt and D. Hawkins have coded several items in the CGRE database. The cooperation of the Computer Center staff is also gratefully acknowledged.

REFERENCES

1. E. E. Tratras and K. J. Johnson, "Computer-Assisted Instruction in Chemistry at Pitt," Proceedings of the Conference on Computers in Chemical Education and Research, DeKalb, Illinois, July, 1971, p. 4-47.
K. J. Johnson, "Curriculum Development Through Computer-Based Materials Graphics, Tutorials and Programming Languages," *ibid.*, p. 5-6.
K. J. Johnson, "Computers and Chemical Education at Pitt," Proceedings of the Conference on Computers in the Undergraduate Curricula, Dartmouth College, June, 1971, p. 130.
2. C. S. Ewig, J. T. Gerig and D. J. Harris, "An Interactive On-Line Computing System as an Instructional Aid," *J. Chem. Ed.*, 47, 97 (1970).
W. J. Karplus, (Ed), *On Line Computing*, McGraw Hill, 1968, pp. 131-78, 303-24.

3. J. A. Pople, W. G. Schneider and H. J. Bernstein, High-Resolution Nuclear Magnetic Resonance, McGraw-Hill, 1959, pp. 218-224.
4. K. J. Johnson, "Numerical Methods in Chemistry--A Computer Applications Course," J. Chem. Ed., 47, 819 (1970).
5. F. Prosser and J. W. Moore, "Computer-Generated Repeatable Tests in Chemistry," Proceedings of the Conference on Computers in Chemical Education and Research, DeKalb, Illinois, July, 1971, p. 9-26.



Jim Durford

PERSPECTIVE #1 by Jim Durford
Problem: Diminishing Polygon Forms

INTERFACING STUDENTS AND COMPUTING THROUGH
UNDERGRADUATE CHEMICAL RESEARCH

L. Chopin Cusachs and Lee P. Gary, Jr.
Loyola University
Telephone: (504) 866-5471

Joyce H. Corrington
Xavier University
New Orleans, Louisiana
Telephone: (504) 486-7411

ABSTRACT

Computing as an avenue to early undergraduate research participation has been realized in three different ways: (1) a formal course in programming and numerical analysis, (2) participation in computer oriented summer research projects, and (3) academic year participation in undergraduate research seminar courses. All three avenues have proved effective in an informal program involving Xavier, Loyola, and Tulane University students and faculty. Benefits to the students include the emotional and intellectual satisfaction of accomplishing useful work, the discovery of talents and development of career interests, and co-authorship of respected original research papers.

Introduction

Early involvement in research by undergraduates is easy when the attraction and teaching potential of the computer is utilized. The possibility of using the computer to reduce the labor of conventional calculations is well recognized, but the practicality of involving students at the early undergraduate level in projects where the problem itself is computational, such as molecular quantum mechanics or crystal structure determination, is little appreciated. The organization of formal introductory computing courses to encourage a research attitude not only provides a more interesting experience for the students, but incidentally relieves the instructor of much drudgery. Both summer and academic year undergraduate research programs organized around computational research have attracted strong student interest and proved rewarding to both unusually gifted and more ordinary students. The following sections consider introductory computing course organization and attitude, as practiced at Tulane and at Xavier by two of us, and the summer and seminar undergraduate chemical research programs as developed over the past few years at the three schools, Loyola, Tulane, and Xavier and Loyola Universities.

Formal Introductory Computing Course

First contact with the computer for many students is the course "Computer Analysis in the Physical Sciences" taught by one of us (LCC) for a number of years at Tulane, and its counterpart at Xavier, "Introduction to FORTRAN Programming." These courses are adapted to the facilities available at Tulane (an IBM 7044) and Xavier (an IBM 1130). Each is open to all students. At Xavier physical science and mathematics majors are required to complete the course.

We regard FORTRAN as a language to be learned by use, since a working or speaking rather than a merely reading knowledge is of value. At the beginning we and the students discuss familiar topics in computation, writing and submitting programs to the computer with the first meeting. As the semester progresses, the fluency of the students increases and the material presented is less and less likely to be familiar to the student, but increasing expertise in the language makes the gradual shift of effort from communicating with the computer to anticipating sources of error seem natural.

A full outline of the course "Computer Analysis in the Physical Sciences" is included in Appendix A. The basic attitude is to start with input/output and progress to numerical methods of increasing sophistication. Each student, in consultation with the instructor, selects a term project, and devotes approximately a quarter of the course effort to it. We have allowed a number of students to take a second semester as seminar or independent study. At the end of the first semester, the average student has a functional command of FORTRAN and is capable of writing programs in his work or research and usually even of adapting large or complex programs from one type of computer to another. Since the large programs of quantum chemistry and X-ray crystallography are normally run in a batch job environment and the available larger computers only operate in that mode, we adopt the batch job attitude from the beginning.

Originally, the orientation of the course to batch job mode was simply making a virtue of necessity. Once we accepted the necessity, we began to teach in ways that utilized the inevitable turn-around time to encourage care, checking, and a measure of forethought. We

believe it is much easier for a student trained to think his programming in this way to adapt to the convenience of terminals, higher level programming languages, etc., than the reverse.

The variety and the quality of the term projects completed shows the considerable potential of this technique for involving students, even beyond the physical sciences students for which it was designed. Project subjects ranged from the obvious, such as automated grading programs and ecological dynamics to outside interests, such as one to transpose music and another to devise minimum-travel bussing plans for a hypothetical public school district.

We followed the suggestions of Mrs. Mary Irvin of the Tulane Computer Laboratory and of Wilson T. Price, author of "Elements of Basic FORTRAN IV Programming," beginning with input/output to place students on the computer in the first class period. A pleasant consequence of this approach is that the papers we received for grading are almost entirely computer output, which is much more readable than our own handwriting, at least. This also made it possible to adopt the policy that a student program that is acceptable to the computer, i.e., produced no error messages, and operated correctly with the data supplied by the instructor, is correct for grade purposes.

Undergraduate Research Programs

The first undergraduate research student in what was to become our inter-university group sought out one of us out of curiosity about the computer. The computer in the area at that time was an IBM 1410 at Tulane, and this student quickly mastered its FORTRAN capabilities and proceeded on to its assembly language and a few tricks like simulating a chain monitor by placing several absolute programs on a tape in succession and loading from the console with an assistant at the tape drive. The fear that such familiarity with the computer might drive students out of the labs quickly dissipated when this first student and his successor went on to graduate school in pharmacology and physiology, respectively, preparing experimental dissertations, though it must be admitted that the first attached a mini-computer to his apparatus to control the instrument and collect data more efficiently.

The next student was encountered in his freshman year as work-study labor for a funded project. It was noted that he began to catch errors in coding assigned to him to keypunch; he had become curious enough to take an informal, non-credit FORTRAN course at the Tulane Computer Laboratory. He collaborated in the writing of programs of increasing complexity, achieving co-authorship of papers in quantum chemistry when he discovered his calling: crystallography. At last word he had completed his doctorate and was working on structure determination on some biological systems. At this point it became apparent that undergraduates could do useful research in quantum chemistry and actual recruiting of research students was begun.

There are two clear advantages to doing research in theoretical chemistry and related fields with undergraduates. First, they are not convinced that the subject is too difficult to master with a little effort. Secondly, it turns out that programming and carrying out calculations is a good way to learn the theory. Particularly for the curious student, the idea of finding out what happens when a particular term is left out or approximated in one way or another by actually doing the calculation several ways has genuine appeal. For the research director on a tight budget, they can also be inexpensive-but productive. There is also a potential for encouraging graduate students to greater effort, particularly when an undergraduate known to be only a B+ student in other courses is able to lecture in a graduate course, citing his own work once or twice for good measure. The effect at Xavier University has been to make Chemistry the largest departmental computer user accounting for 30% of all academic use.

Formal avenues to undergraduate research participation include the standard "senior" research seminar courses encouraged by most departments when not specifically required, and summer programs such as the NSF Undergraduate Research Participation Program.

The standard senior research seminar course is probably the least effective. By the senior year it is usually too late to do much for the undergraduate. Getting started in research takes most of a year, and the senior tends to be already looking beyond graduation to graduate school, job, or service elsewhere. The freshman or sophomore is better able to spend the long hours required to get started without hurting his regular academic performance. We find the grades of freshmen or sophomore research students in unrelated courses tend to rise when they get involved in research-while senior research generates distractions. The senior may actually be learning much that is valuable, but he usually is too rushed to produce results commensurate with the time his supervisor invests. When the student starts early in his career, he has time to learn from others ahead of him and to pass on some of his knowledge to his juniors. The feeling of participation in a small group with a real purpose often helps the shy or uncertain student to mature socially as well as intellectually. For these reasons we stress the desirability of introducing students to research early in their undergraduate years.

However, with the assistance of previously accumulated subroutines and calculation programs even seniors with no previous computing or research experience can complete a worthwhile projection a single term. One of us (JHC) supervised two such students in the spring of 1971, and each student completed and presented a paper on his work at a local American Chemical Society meeting.

The intensive summer programs are particularly valuable but depend usually on outside funding for student stipends to attract participants.

Professor L. P. Gary of Loyola, under an NSF-URP grant, supervised two students and Professor Joyce H. Corrington of Xavier, under an ACS-PRF Type B grant, supervised six students in the summers of 1970 and 1971.

In addition to the obvious advantages of full time effort in research, the summer research student can make a commitment for a specific period in an area of research and change in the following year without having to explain to a disappointed supervisor why he really prefers, for instance, crystallography to quantum chemistry. The practice, in our group, of encouraging students to try other research areas and other institutions for a summer to sample some of the other interests available has produced some switches out of our research group but provided a better overall educational opportunity for the student.

Project (Appendix B) offers to summer and to academic year research students range from the development of better programs and computational algorithms to the more applied study of families of molecules or reaction systems to determine how well the methods account for the physical and chemical properties of the systems. As examples, Benes Trus and C. W. McCurdy worked on approximations and computer programs for molecular orbital calculations and the computation of transition moments and transition energies for electronic spectra. It is interesting that McCurdy discovered a manipulation to simplify calculation of electronic spectra that was almost simultaneously noted by two established investigators at distant institutions. As programmers there is no reason to believe that undergraduates should be inferior to their faculty supervisors if they get enough practice. We should stress that participation in undergraduate research in our group implies no commitment to our field after graduation; an example is the work of J. B. Florey on pseudorotation in PF₅, which did not conflict with his plans for medical school, preparing for a career in psychiatry. A selected list of publications involving major contributions by undergraduate research students is included (Appendix C).

The one area in which undergraduates have been of little help to us is in writing the final manuscripts for publication. Had this not been the case, the list of publications would be much longer. However, some of the students did prepare an undergraduate thesis in fulfillment of the requirements for an Honor Degree.

They have actively prepared slides and presented papers at local ACS meetings. At the 1970 New Orleans ACS Meeting in Miniature students collaborating with the authors presented ten of the fifteen papers accepted for the Physical Chemistry Session. The papers are listed in Appendix D.

ACKNOWLEDGMENTS

The work discussed in this paper was supported in part by: The Xavier Computer Center, The Tulane Computer Laboratory and the Loyola Computer Center, and by grants from the American Chemical Society Petroleum Research (Grant PRF-5056), the Edward G. Schlieder Foundation and the National Science Foundation.

APPENDIX A

"Computer Analysis in the Physical Sciences"

- A. Concept of Course: FORTRAN is a Language, Computer Techniques the Subject.
1. The course builds on one year of college mathematics and one year of a course in natural science.
 2. Batch Mode, typical of actual installation practice, is employed.
 3. Assignments stress achievement of a functioning program, rather than caution to avoid errors penalized in grading.
 4. Individual projects challenge gifted students while the average one solidifies his competence.

5. A language is taught by use, a technique by analysis and experimentation.
 6. Course parallels actual use of computer for real problems.
- B. Organization of the Course: sequence of assignments.
1. Two lecture/laboratory meetings per week, plus instructor's availability outside of class.
 2. Material for grading, with one exception, is computer output; fast FORTRAN (WATFOR) compiler keeps cost of this acceptably low (less than \$20.00/student at Tulane; \$50.00/student at Xavier).
 3. Early establishment of Student-Compiler dialogues helps student know where he stands, "If it works correctly with the instructor's test data, it is correct." Workable with inexperienced instructor, "Try it and see if it works."
 4. Stress on assignments due on time over the semester minimizes conflicts with traditional courses at exam times.
- C. Sample Schedule of Assignments.
1. Key punching and Control Cards (ungraded), READ, WRITE, FORMAT.
 2. Elementary Arithmetic in FORTRAN (ungraded) Integer, Real Arithmetic.
 3. Construction of a Table of Values of a Polynomial, DO Loop, Arithmetic Statement Function.
 4. Roots of a Polynomial by Half-Interval Method (Regula Falsi) IF statements, Indexing, Modulo arithmetic, Arrays for data storage.
 5. Roots of a Polynomial by Newton-Raphson method, Convergence of iterative processes, Debugging tactics.
 6. Roots of a Transcendental Equation (optional), Library functions.
 7. Vectors & Matrices (graded hand assignment), fundamental definitions and manipulations.
 8. Vectors & Matrices, manipulation in FORTRAN (ungraded).
 9. Linear Equations, Gauss-Jordan elimination method, Indexing, More indexing.
 10. Least Squares, Subroutines, COMMON.
 11. Numerical Integration, Simpson's and Trapezoidal Rules, Gaussian Quadrature Formulas, Rounding vs. truncation error.
 12. Term project: Valued at 3 regular assignments.
 13. Optional, Ungraded, Eigenvalue Problems in Non-Orthogonal Basis, Nonlinear Regression.
 14. Optional ventures in COMPLEX, and Double Precision Arithmetic, or Boolean Algebra--it the compiler is up to it!
- D. Precautions & Variations.
1. The ringer with some programming experience--use as tutor at beginning if cooperative.
 2. The gifted student--encourage side explorations and refinements.
 3. The slow student--leave open a straight, simple road.
 4. Selection of Project "by negotiation."
 - a. Established interest of student, research or other.
 - b. Programming ability of student.
 - c. Scientific or other talents of student.

5. Time requirements - at Tulane, typically about 7 hours of IBM 7044 time for 25 students. At Xavier about 30 hours of IBM 1130 time for 110 students.
6. Optional Second Term
 1. Seminar organization...the student is ready to read.
 2. Theme...something like differential equations and the atomic self-consistent field problem.
7. Museum or Student Programs and Projects (Appendix B)

APPENDIX B

Sample of Student Term Projects

1. Pupil Assignments to Minimize Busing (Social Sciences): A model of a school district with non-uniform racial and socio-economic populations and school building distribution. Compute assignments to minimize the overall busing required for balance.
2. Titration Simulation (Chemistry): Specifying an amount of base in known volume of solution and concentration and drop size of standard acid, simulate the addition of successive drops of acid to base, computing pH and other ion relations, does not attempt to reduce to quadratic or linear approximations.
3. Comparison of Stability and Efficiency of Numerical Treatments of Differential Equations: Compares several popular numerical techniques with respect to stability and computational efficiency.
4. Long Term Regulation of Extracellular Fluid and Arterial Volume by Control of Urine Output (Physiology): Digital implementation of a model of fluid balance factors in a living system. Useful to test responses to various disorders or insults.
5. Statistical Package for Quantitative Chemistry Lab (Chemistry): A simple but useful statistical program package for the more frequent laboratory calculations in quantitative analysis.
6. Nuclear Shape Calculations (Physics): Calculations using modifications of fluid drop model to estimate shapes of heavy nuclei of atoms.
7. Heat Transfer in a Square Pipe (Physics): The heat transfer in an ordinary circular pipe is a standard textbook problem. A square pipe is interesting, useful, and messy enough to be natural for a computer.
8. Prediction of Power Diffraction Patterns (Crystallography): Computer prediction of power X-ray diffraction patterns for a specified assumed crystal structure.
9. A 3-dimensional Plotter Program for the Line Printer: Attempt to represent 3-dimensional data on a two-dimensional line printer.
10. Solution of the atomic radial Schrodinger equation with Thomas-Fermi or Thomas-Fermi Dirac Potentials (Physics).

APPENDIX C

Typical Publications Involving Major Contributions by Undergraduates

1. "Pictures of Molecular Orbitals" H. B. Becker, H. E. Longenecker, Jr., and L. C. Cusachs, Communications of the Association for Computing Machinery 8, 581(1965).
2. "Calculation of Molecular Transition Moments" L. C. Cusachs and B. L. Trus, Journal of Chemical Physics 46, 1532 (1967).
3. "Selection of Molecular Matrix Elements from Atomic Data, III." L. Cusachs and J. R. Linn, Jr., Journal of Chemical Physics, 46, 2919 (1967).

4. "Overlap Matched Atomic Orbitals," L. C. Cusachs, D. G. Carroll, and S. P. McGlynn, B. L. Fius, Int. J. of Quantum Chem., 15, 423 (1967).
5. "Thermodynamic Functions from Kinetic Data: A Nonlinear Approach," R. K. Nichols, T. F. Pagley, and L. C. Cusachs, Student Members Bulletin, AICh E., 8, 39 (1967).
6. "The Mechanism of the Hydrogen-Iodine Reaction at Low Temperatures," L. C. Cusachs, M. Krieger, and C. W. McCurdy, J. Chem. Phys., 49, 3740 (1968).
7. "Conservation of Molecular Orbital Configuration in Chemical Reactions," L. C. Cusachs, M. Krieger, and C. W. McCurdy, Int. J. Quantum Chem., 15, 67 (1969).
8. "The 4s Orbital of Sulfur," L. C. Cusachs, D. J. Miller and C. W. McCurdy, Jr., Spect. Ltrs. 2, 141 (1969).
9. "Analysis of Hydrogen Bonding in Ammonia Dimers Using an LCAO-MO Scheme," H. S. Aldrich, H. C. Hasenkampf, H. J. Ladef, L. C. Cusachs, and L. P. Gary, Bull. Am. Phys. Soc., 15, 1335 (1970).
10. "Electronic Effects in the Structure of Some Silver Iodide Complexes," by H. B. Jonassen, C. W. McCurdy, Jr., L. C. Cusachs, G. B. Ansell, and W. G. Finnegan, NWTC-TP 4996 (August, 1970).
11. "Dipole Moments and Orbital Energies from ARCANA: A Semiempirical Molecular Orbital Calculation Program," J. H. Corrington, H. S. Aldrich, C. W. McCurdy, and L. C. Cusachs, Int. J. Quantum Chem., 55 in press.
12. "Simplification of the RPA Secular Equation," by C. W. McCurdy and L. C. Cusachs, J. Chem. Phys. 54, (1971).
13. "Semiempirical Molecular Orbital Calculations: Pseudorotation in Phosphorus Pentafluoride," by J. R. Florey and L. C. Cusachs. J. Amer. Chem. Soc.

Note: Underscore denotes student.

APPENDIX D

Student Generated Research Papers for ACS Meeting

1. "Dipole Moments and Orbital Energies from Arcana," H. S. Aldrich, C. W. McCurdy, and L. C. Cusachs.
2. "Simplified Calculations of Atomic and Diatomic Integrals," Jennifer Christopher and Joyce H. Corrington.
3. "An LCAO-MO Study of the 1, 2-Dihalogenoethylene Series," L. M. Keefer, L. P. Gary and L. C. Cusachs.
4. "Molecular Orbital Studies of Neurotransmitters," W. D. Burton, Jr., and Vernon B. Haarstad.
5. "Electron Bonding in Azomethane," H. C. Hasenkampf and S. T. Kent.
6. "Structure-Bonding Studies of Hemicholinium HC-3," Andrew G. Pattamana and Joyce H. Corrington.
7. "Application of the Random Phase Approximation to the Semiempirical Molecular Orbital Calculation of Electronic Spectra." C. W. McCurdy and L. C. Cusachs.
8. "A Study of the Structure-Activity Relationships of 1,4-Benzodiazepines Based Upon Molecular Orbital Calculations," M. F. Murphy, R. Guerrero-Figueroa, D. Galland, L. C. Cusachs.
9. "Semiempirical MO Calculations of Model Systems for Drug Receptor Complexes: Methylamine and Acetic Acid," Milton Coleman and Joyce H. Corrington.
10. "Triplet-Triplet Interaction Between Carcinogenic Aromatic Hydrocarbons," H. S. Aldrich and L. P. Gary.

Note: Underscore denotes student.

A CAI PROGRAM FOR AROMATIC ORGANIC SYNTHESIS WRITTEN
IN TIME SHARING FORTRAN

Ronald D. Crain
University of Kansas
Lawrence, Kansas 66044
Telephone: (913) 864-4097

A beginning student in organic chemistry usually gets very little understanding of syntheses until the benzene ring and its chemistry is introduced into the lecture material. At this point the students begin to learn that all one does is introduce, alter, and remove groups from a fixed ring system according to certain defined rules. In fact, this type of synthesis is referred to as "tinker toy" because the ring can be viewed as a constant frame of reference.

Problems are normally assigned for homework and the lecturer covers in class the general approach that should be followed by the student. The answers finally get posted and the student has a chance to discover which ones he missed. The better students are often able to immediately see where they went wrong but most students wonder why a particular synthesis was incorrect and what step was wrong. Unfortunately, the answer can often be that they were not all wrong but that they simply had not taken the standard route. This can be clarified for the student if he talks with an instructor. In a large institution this is quite difficult because there are too many students and not enough time to help everyone requesting the additional help.

One of the obvious approaches for such logistical problems in teaching is to develop a CAI program to handle the majority of the student's need for confirming which synthetic steps, and in what sequence, are feasible for making a compound. It would be far easier to write a program which requires that a student begin with the starting material and proceed stepwise to the product. However, that is not the way an organic chemist would approach the problem. One starts at the end and does the last step first. Occasionally you jump to the beginning and take the starting material one step forward and compare it with possible routes to the other end. Thus, to be pedagogically correct an organic synthesis CAI program would have to have this versatility.

There have been several programs produced for synthesizing organic benzene compounds. Dr. Stanley Smith, University of Illinois, has reported[1] on the use of their PLATO system for such purposes and Dr. L. B. Rolewaid and coworkers[2] at the University of Texas have several completed programs in this area. To the best of the author's knowledge none of these programs allow a student the privilege of starting anywhere along a reaction path.

Such a program has been written in time sharing FORTRAN for use with the HW-6J5 computer at the University of Kansas which attempts to meet with this requirement. In addition, this program should have a relatively easy transferability to other computer systems since standard teletype or CRT terminals are used and the programming language is slightly modified FORTRAN IV.

Program Use

A student is given a sheet of paper which describes how to call the program at the terminal, a list of reagents which can be used, and a list of substituents which could be attached to the ring (or removed). The present program can utilize up to three substituents on the ring at any given time (see Table 1), thirteen different substituents, and 24 reagents. The chemistry of the aliphatic side-chains, other than oxidation, is not included in the program.

Problems can be assigned by the instructor or the student may use those in the text or make up his own synthesis problem. Upon entering the program (Figure 1) the benzene ring is printed. The student may either start with the benzene or add substituents, up to three, for his first reaction. Suppose he added a CH₃ in the 'one' position. The structure of toluene is printed and the student is asked to enter the reagents to be used. The major product from the reaction is printed and the student has the option to continuing the synthesis to a third compound, getting a new starting material, or quitting. Let's assume that he has nitrated and now has 4-nitrotoluene as a product (Figure 2). If potassium dichromate and sulfuric acid are chosen, 4-nitrobenzoic acid's structure is printed. If tin and hydrochloric acid is used the structure for 4-amino-toluene (p-toluidine) is printed at the terminal. Reaction of the amine with an oxidizing agent will result in a message being returned that this type of reaction gives a terrible mixture. The student is then offered a chance to try some other reagent(s).

There are features of this program which have helped the student become familiar with the techniques needed for a successful synthesis problem. The presentation of all of the reagents (tools) in one place allowed the student to get a different perspective on the use of these compounds in synthetic work. It has helped to reorient their thinking from the concept that each step is a new one requiring entirely different and unrelated chemicals (a common misconception

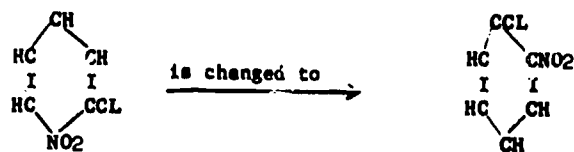


FIGURE 4. Changing a structure stored in BENZ.

Reagents Available		
HNO ₃	FeBr ₃	K ₂ Cr ₂ O ₇
H ₂ SO ₄	FeCl ₃	KMnO ₄
HCl	AlCl ₃	Mg
H ₃ PO ₂	NaNO ₂	CO ₂
Br ₂	CuBr	Ac ₂ O
Cl ₂	CuCl	AcCl
Fe	CuCN	
Zn	KI	
Sn	NaOH	

Allowable Groups			
CH ₃	OH	Br	NO ₂
NH ₂	OAc	Cl	N ₂ Cl
NHAc		I	CO ₂ H
Ac			CN
			SO ₃ H

TABLE 1. Reagents and substitutes for Aromatic Synthesis program.

with beginning students) to be memorized. The students have the freedom of checking out a given compound with a given set of reagents without even having a synthesis in mind.

The not-so-good features are mainly due to the limitations of a standard teletype terminal. The printout rate is 10 characters per second thereby making the construction of a two-dimensional molecule a little bit lengthy (slow). The steps in the synthesis are also printed in a vertical fashion rather than the standard approach used by text, teacher and student: (horizontally). At the present time no attempt has been made to save, upon command, a given stop and then print out the entire completed synthesis when the student is satisfied that it is correct. There is also no attempt to check the last compound produced in the synthesis to see if it agrees with what the student set out to make. These features are nice but the program could become quite large for many systems and then a loss in transferability occurs. (They are being included, when time permits, for our own internal use).

One slightly bad pedagogical item should also be mentioned since the program as presently constructed makes no mention of a chemical reality. Quite often there are two possible products obtained from a reaction. This program gives only the major product. The inclusion of the student option of picking one of two possible products is being studied but it may well be somewhat dependent upon special display units (CRT) which in turn would make the transferability of the program more difficult unless the user also had a similar terminal.

The internal working of the program is guided by one data array (two-dimensional) in which the structure of the benzene ring is stored in character form (Figure 3). Alterations to this structure are dictated by the reagents selected and any other groups that may be present. For example, if the reagents selected dictate the placing of a nitro (NO_2) group on the ring the NO_2 is read from the data array FUNC into the benzene data array (BENZ) starting at a specified point in the BENZ array. The same thing could have been accomplished by having the student select a nitro group and request that it go to the 'one' position. A simple print subroutine then prints out what is stored in BENZ as a two-dimensional molecule. The BENZ and FUNC arrays are set up in ASCII A1 fields but that is not necessary. Routines have been done where the same type of thing can be done using A2 or A4 fields.

There is a second routine which can check the BENZ array for the type of substituent and where it is located. If there is nothing on the 'one' carbon atom (see Figure 4) other than a hydrogen atom and a group is found on the second or third carbon atom then the program redoes the structure so that the group appears on the first carbon. This was included in the program to reduce the actual number of statements to handle the rather large variety of possibilities. If one has a compound like 1,2-dichlorobenzene it is actually the same as 3,4-dichlorobenzene (or 2,3- or 4,5-, or 5,6-, or 1,6-). Thus the variation is reduced by a factor of six in this one instant. In the latter cases they would all be translated in the BENZ array as 1,2-dichlorobenzene and printed as such.

The same routine can also determine the branching in the program where the chemistry of (1) benzene; (2) monosubstituted benzenes; (3) 1,2-disubstituted; (4) 1,3-disubstituted; (5) 1,4-disubstituted; (6) 1,2,3-trisubstituted, etc. This allows the student to build a benzene ring with up to three substituents on it or actually make the compound during a synthesis and this program would check out the molecule, shift the groups if necessary, and branch to the area where the chemistry of that type of substituted benzene is covered.

AN INTERACTIVE TIME-SHARING BASIC TUTORIAL PROGRAM SEQUENCE
IN INTRODUCTORY ELECTROCHEMISTRY

Alfred J. Lata
University of Kansas
Lawrence, Kansas 66044
Telephone: (913) 804-4054

In an effort to assist our General Chemistry students in and through those areas of the course which are perennially difficult, we investigated the possibility of utilizing Computer Assisted Instruction (CAI). However, the University of Kansas does not have a CAI language such as Coursewriter or PLANIT, or a system such as PLATO available. Efforts to convince our Computation Center staff to adopt and/or develop a CAI language have fallen on sympathetic ears. The Center's feeling, however, was that with only one computer available (a Honeywell 635) and with our software facilities, a lengthy period of time would be necessary to adopt and modify one of the existing CAI languages, and the amount of memory necessary would mean that the computer would have to be dedicated to CAI, to the exclusion of general use. We were advised that if we wished to do any CAI work within the next one to two years at least, we would need to use existing facilities.

We were, and are, therefore in the same situation as many other schools throughout the country. We soon recognized the fact, however, that if we were able to write effective CAI programs in TSS BASIC and/or TSS FORTRAN, these programs could be used by other schools on their own computers without the need of their having a CAI language available. What we had thought a handicap, has proved to be instead an advantage, in that these programs will be readily transferable.

We have now written and tested CAI programs in TSS BASIC and TSS FORTRAN. The programs probably take longer to write in these languages than in a CAI language, but they also work effectively with students. We have used both CRT and teletype for output, the latter being more desirable since the student has hard copy for review. The fact that there are no lower case characters or subscripts is a minor problem that is easily overcome by the students. As evidence of their transferability, several of our programs have been taken to other schools and with minor modifications have run on their computers.

To show a portion of what we have achieved, I wish to discuss a group of computer assisted instruction programs written in time-sharing BASIC covering topics in beginning Electrochemistry commonly discussed in a General Chemistry course. There are seven programs:

1. REDOX is a program designed to teach students the ion electron method of balancing of redox equations assuming no previous experience.
2. REDOX2 allows the student to practice balancing redox reactions by the ion-electron method at either, or both of two more advanced levels than the previous program.
3. CELLCALC: the student calculates the potential of a cell from two reduction potentials. All concentrations are 1 molar.
4. STDCALC allows the student to find the potential of a half cell given the whole cell potential and the potential of a reference electrode to which the half cell in question is coupled.
5. NERNST: the student determines the potential of a half cell if the concentrations of the ionic species in the half cell are other than 1 molar.
6. CONCPOT has the student find the concentration of an ionic species given the potential of the half cell.
7. COMPCALC is a program in which the student is given the potential of a cell, composed of a reference electrode and a half cell with either a saturated solution or a solution of a complex ion whose K_{sp} or K_{diss} the student determines.

The random number generator function was used in all programs to select data, both equations and numbers, for the student's problem; this means that the probability of two students getting the identical problems or equations is small. There are few fixed answers in any of the programs, answers being calculated by the computer and the student's answer then compared with the calculated result. The student is asked to respond to a question or problem. If his answer is correct, he may choose to go on to another problem or the next step. If the

answer incorrect, he is either given a diagnostic response and/or told to try again, or he is asked to respond to a question dealing with the immediately previous step in the calculation, by which it may be possible to determine the student's error and give him a diagnostic. This retrograde questioning continues until the student's error is located.

The student has the flexibility to get out of a problem by responding "99" or "out" to a question, or to seek assistance by simply typing "help" to find out the answer to the previous question; this is done by a subroutine which examines each answer for one of these responses. Because of a floating point conversion problem in comparing student numerical responses to computer generated mathematical solutions, each student mathematical answer is examined to determine whether it is within an allowable percentage range; this percentage existing in the program for each calculation.

I wish to discuss the programs briefly, stressing particularly the programs used first in the program sequence--REDOX and REDOX2, which are non-mathematical in nature; CELLCALC will be used to illustrate the remaining programs to give an idea of how mathematically oriented programs function in this group. All programs were originally written as integral programs to stand alone; however, several subroutines are common to several programs and are now called up from the file when needed.

The initial problem to be solved in developing the program REDOX, was to code the data statements in order to have available for each half reaction the reacting species, products, the number of hydrogens and oxygens in each species, the number of electrons for the half reaction, and whether the reaction takes place in acidic or basic medium. The reduction half reaction



is coded in the following fashion for the BASIC data statement.

```

1 H3ASO4 4 3 2 1 HASO2 2 1 0.56 1
A      B   C  D E F   G   H I   J  K

A = # of H3AsO4 in equation      F = # of HASO2 in equation
B = oxidizing agent              G = reducing agent
C = # of O's in oxid. agent      H = # of O's in reduc. agent
D = # of H's in oxid. agent      I = # of H's in reduc. agent
E = # of electrons                J = Potential of half cell
K = 1 for acid, 0 for neutral, -1 for basic

```

The data available in the program are 34 reduction half reactions (including two organic reductions) arranged in order of decreasing reduction potential. Two half reactions are selected by use of the random number generator. The pair is examined to check whether acid-base compatibility of the two reactions is observed by comparing the values K, the last value in the data statement. If the half reactions are incompatible, one of the reactions is replaced by another randomly selected half reaction. In our current file of 34 half reactions, 22 are in acid solution and 12 are in base; this gives 231 possible whole cell equations in acid, and 66 possible in basic solution. If all half reactions were in acidic solution, there would be 561 different possible whole cell reactions that could be attempted by the student. After the student signs on and selects program REDOX, he is given some introductory information and then the unbalanced equation is given him.

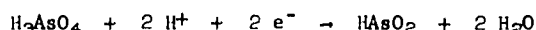
```

MNO4- + FE+2 → MN+2 + FE+3   IN ACID SOLUTION
or
BRO- → BR- + BRO3-   IN BASIC SOLUTION
THIS IS A DISPROPORTIONATION REACTION
or
CLO4- + CLO2- → CLO3-   IN BASIC SOLUTION

```

It is only in the initial equation that the disproportionation or the single product is shown. You will note that one of the reduction half reactions has been reversed to show that oxidation is taking place. This is performed internally within the program. Shown here is a portion of the beginning of the program, remember that this is an introductory elementary balancing of the redox equation.

The program is capable of handling equations in either acidic or basic medium. The number of oxygens, hydrogens, and waters needed in the equation are calculated from the number of oxygens and hydrogens in the oxidized and reduced species: each excess oxygen on the left requires two hydrogen ions and forms one water in acid solution; or in basic solution, each excess oxygen on the left requires one water and forms two hydroxide ions. In acid solution, if there are more hydrogens in the reacting species than in the product, this decreases the number of protons needed on the left, e.g.,



There are two excess oxygens on the left [# of O's in oxidized form (4) - # of O's in reduced form (2)], and the two excess H's on the left [calculated in the same manner, (3-1)], reduce the number of protons needed on the left to two. Similar calculations are performed by the computer for reactions in basic medium. A portion of the program is shown here:

LET'S BALANCE THE EQUATION:

CR2O7-- + HASO2 → CR+3 + H3ASO4 IN ACID SOLUTION

WHAT SUBSTANCE IS BEING REDUCED?

?CR2O7--

WHAT SUBSTANCE IS BEING OXIDIZED?

?HASO2

GOOD, JOHN

FOR THE REDUCTION HALF REACTION CR2O7-- → CR+3 YOU WILL NOTICE THAT ONE CR2O7-- GIVES 2 CR+3 TO BALANCE THE NUMBER OF ATOMS OF THE ELEMENT BEING REDUCED

WHICH SIDE HAS MORE OXYGENS?

?LEFT

CORRECT, JOHN! HOW MANY MORE ARE THERE ON THE LEFT SIDE?

?14

TRY AGAIN, JOHN

?7

RIGHT, JOHN!

HOW MANY WATER MOLECULES WILL THE OXYGEN FORM?

?7

VERY GOOD, JOHN 1 CR2O7-- → 2 CR+3 + 7 H2O THIS WILL REQUIRE HOW MANY H+'S ON THE LEFT?

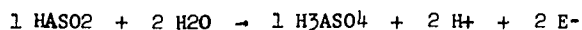
?7

NO, REMEMBER THAT EACH WATER GIVES TWO H+ IONS.

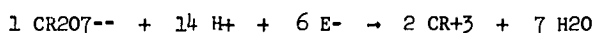
TRY AGAIN

?14

The student goes through both the oxidation and the reduction half reactions in this manner and then he determines the least common multiple of electrons for both half reactions



NOW LET'S COMBINE THE HALF REACTIONS. THE REDUCTION HALF REACTION WAS:



WHAT IS THE LOWEST COMMON MULTIPLE NUMBER OF ELECTRONS COMMON TO BOTH EQUATIONS?

?12

NO, JOHN, WHAT IS THE LEAST COMMON MULTIPLE OF 2 AND 6?

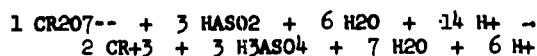
?6

RIGHT! HOW MANY OF THE REDUCTION HALF REACTIONS DO YOU NEED TO TAKE UP 6 ELECTRONS?

?1

couples the half reactions, and determines the number of each species on both sides of the equation,

THE SUM OF THE TWO HALF REACTIONS IS THEN:



WHICH SIDE HAS MORE WATER MOLECULES?

?RIGHT

HOW MANY MORE WATER MOLECULES?

?1

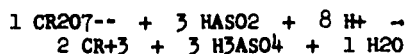
GOOD. AND WHICH SIDE HAS THE GREATER NUMBER OF H+ IONS?

?LEFT

HOW MANY MORE H+ IONS?

?8

GOOD. NOW THE EQUATION SHOULD READ:



In the program REDOX2, a different coding is used for the equation, and this coding is used in all subsequent programs except COMPCALC.

For reactions in acid solution



the coding is:

1	H3ASO4	2	2	1	HASO2	2	0.56	1	1	1
A	B	C	E	F	G	H	J	K	L	M

For reactions in basic medium



the coding is:

2	ClO-	2	2	1	Cl2	4	0.40	-1	1	2
A	B	C	E	F	G	H	J	K	L	M

A = # of ClO-'s in equation F = # of Cl2 in equation

B = oxidizing agent G = reducing agent

C = # of H+ if acidic H = # of H2O if acidic
of H2O if basic # of OH-'s if basic

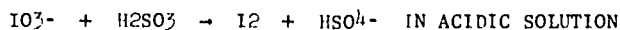
E = # of electrons J = reduction potential

K = 1 for acidic, -1 for basic

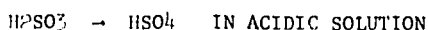
L for oxid. agent	} {	0 for solid,
M for reduc. agent		1 for ion or soluble molecule
		2 for gas

This coding differs from that in the previous program in that hydrogen ion, or hydroxide ion, and water are included in the data for each equation rather than the number of oxygens and hydrogens in each individual species. Also at the end of the data statement a coding for solids, ions or molecules which can vary in concentrations, and gases is included for both the oxidizing agent and the reducing agent. In this program [with two levels of difficulty (1) balancing each half reaction and then the whole reaction and (2) balancing only the whole final reaction] the student determines how many hydrogen ions (or hydroxide ions if basic) and waters are necessary in each equation. A portion of this program is shown below.

BALANCE THE FOLLOWING EQUATION:



BALANCE THE FOLLOWING OXIDATION HALF REACTION



ON THE LEFT SIDE: HOW MANY H₂SO₃-?

?1

HOW MANY WATERS?

?1

HOW MANY H⁺'S?

?0

ON THE RIGHT SIDE: HOW MANY HSO₄⁻-?

?1

HOW MANY H₂O'S?

?0

HOW MANY H⁺'S?

?4

HOW MANY ELECTRONS?

?3

THE NUMBER OF H⁺'S ON THE RIGHT SIDE IS WRONG.

THE NUMBER OF ELECTRONS ON THE RIGHT SIDE IS WRONG.

BALANCE THE EQUATION AGAIN AND WE'LL TRY AGAIN.

(The equation should read $\text{H}_2\text{SO}_3 + \text{H}_2\text{O} \rightarrow \text{HSO}_4^- + 3 \text{H}^+ + 2 \text{e}^-$)

After each half reaction is worked correctly in this fashion, the student then finds the number of electrons common to both half reactions, couples the two half reactions and then is questioned in the same fashion about the final balanced equation.

NOW THAT YOU'VE DONE BOTH HALF REACTIONS, COUPLE THEM AND I'LL CHECK YOU ON THE BALANCED EQUATION. HOW MANY ELECTRONS ARE COMMON TO BOTH EQUATIONS?

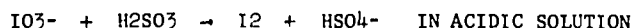
?10

RIGHT, JOHN, NOW CONSIDER THE WHOLE BALANCED EQUATION

ON THE LEFT SIDE: HOW MANY HSO₄⁻-?

In the second level of REDOX2 the student is given only the skeletal equation and must work out the individual half reactions on his own with no computer dialogue and respond only about the final balanced equation; this is evaluated as were the half reactions at the previous level and if a mistake in the number of one of the species is found, the student is so informed and sent back to resubmit the number of each species. An example is shown below.

YOU WILL BE GIVEN A SKELETAL REDOX REACTION TO BALANCE:
BALANCE THE FOLLOWING:



I'LL CHECK YOU ON THE BALANCED EQUATION.
ON THE LEFT SIDE: HOW MANY H₂SO₃?

?5

HOW MANY IO₃⁻?

?2

HOW MANY WATERS?

?0

HOW MANY H⁺'S?

?0

ON THE RIGHT SIDE: HOW MANY HSO₄⁻?

?5

HOW MANY I₂?

?1

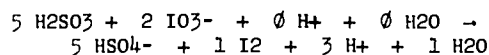
HOW MANY H₂O'S?

?1

HOW MANY H⁺'S?

?3

THAT'S RIGHT, AND THAT GIVES US FOR THE
BALANCED EQUATION



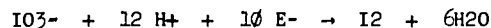
WANT TO TRY ANOTHER ONE?

In program CELLCALC the student is asked to find the potential of a cell made up of two half cells selected at random from the table. The student should have the table of reduction half reactions and potentials at hand since in only one of the three levels of this program are potentials for the half cells given in the output. The concentrations of all species are considered to be one molar. The student may initially choose which of three levels at which he wished to start work; five successful problems at one level advances him to the next higher level where the problems are slightly more difficult. In Level 1 the half cells are given with the potentials:

CONSIDER A CELL COMPOSED OF THE TWO HALF CELLS:
(ALL CONC AS 1 M)



WITH A REDUCTION POTENTIAL OF 0.17 VOLTS



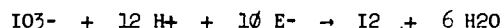
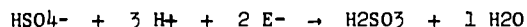
WITH A REDUCTION POTENTIAL OF 1.20 VOLTS

IN ACIDIC SOLUTION

WHAT IS THE POTENTIAL OF THIS CELL?

In Level 2 only the half cells are given, the student must find the potentials from the table:

CONSIDER A CELL COMPOSED OF THE TWO HALF CELLS:
(ALL CONC ARE 1 M)

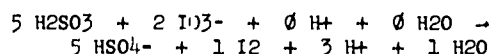


IN ACIDIC SOLUTION

WHAT IS THE POTENTIAL OF THIS CELL?

And in Level 3, the balanced redox equation is given and the student must decide which half cells and corresponding potentials to pick from the table:

A CELL HAS THE CHEMICAL REACTION:
(ALL CONC ARE 1 MOLAR)

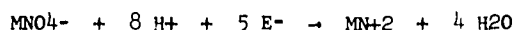


WHAT IS THE POTENTIAL OF THIS CELL?

When the student responds with his answer it is compared to the answer calculated from the data given for the two half reactions, by subtracting the reduction potential of the half cell undergoing oxidation from the potential of the other half cell. If the student's answer does not match the computer calculated answer within allowable limits, his answer is then compared against other possible incorrect answers calculated using the mistakes most commonly committed by students: (1) simply adding the two reduction potentials; (2) reversing the sign of the wrong half cell which will result in a negative potential which is not indicative of a spontaneous reaction; and (3) multiplying the potential by the number of electrons involved in the change. For any other answer, the student is quizzed as to what reduction potential he used for each half cell. If it is found that one of these is in error he is told to go back and do the problem again; if the reduction potentials he used are correct he is then told that he apparently made a computational error and to try the problem again.

The program STDCALC is used to give the student practice in the calculation of a half cell potential from a whole cell potential in which one of the half cells is a reference cell whose potential he knows. The initial question is given to the student in the following form:

FIND THE REDUCTION POTENTIAL OF THE HALF CELL



WHICH WHEN COUPLED TO A SATURATED CALOMEL REFERENCE
ELECTRODE HAVING A REDUCTION POTENTIAL OF 0.249
VOLTS, GIVES A CELL GENERATING 1.45 VOLTS. THE
REFERENCE ELECTRODE IS THE ANODE.

In the problem three things are chosen using the random number generator: (1) the half cell in question; (2) the reference cell; (3) the cell potential. The reference cell is designated as either anode or cathode by comparing its potential with the calculated potential for the half cell in question.

Programs NERNST and CONCPOT utilize the Nernst Equation,

$$E = E_0 - \frac{.059}{n} \log \frac{[\text{RED}]^b}{[\text{OX}]^a [\text{H}^+]^x}$$

Again, the random number generator selects for the student a half reaction with which to work. In program NERNST, randomly selected concentrations are given for the chemical species and the student is asked to calculate the potential of the half cell. In program CONCPOT the student is given a potential for the half cell and asked to calculate the concentration of one of the species [(OX), (RED), or (H⁺)], given the concentrations of the remaining species in the half cell. As in the previous programs, percentage limits are set and possible incorrect answers are calculated and compared to the student's answer, so that if it is in error a diagnostic may be given.

Program COMPCALC combines several of the previous programs the student is asked to determine K_{diss} or K_{sp} given the potential for a cell composed of a reference electrode chosen at random and one of the half cells, containing either a complex ion or a saturated solution. He must go through the process of finding the potential of the half cell; the concentration of the ion in question using the Nernst equation; and, with the concentration of complexing, or precipitating species, the equilibrium constant.

A CELL COMPOSED OF AN ANTIMONY REFERENCE ELECTRODE HAVING A REDUCTION POTENTIAL OF 0.145 VOLTS IS COUPLED TO A ZINC ELECTRODE IN A $\text{ZN}(\text{NH}_3)_4^{++}$ SOLUTION (0.1 M). THE EQUIL. CONC. OF NH_3 IS 2 M AND THE POTENTIAL OF THE CELL IS 1.249 VOLTS. THE REFERENCE ELECTRODE IS THE CATHODE.
WHAT IS THE K DISS OF $\text{ZN}(\text{NH}_3)_4^{++}$?

Conclusion

We believe that we have written educationally effective tutorial programs for electrochemistry in time-sharing BASIC, programs that can be easily transferred to other computers. It is possible to write interactive CAI programs without a CAI system or language. We are currently in the process of converting our BASIC programs to FORTRAN which may allow for still greater transferability. It is our belief that if individuals at various schools write CAI programs in the more common languages and share these programs with other schools, effective libraries of these materials can soon be created for student use at many institutions. We urge you to join us in this endeavor.

MICROMOD: DESCRIPTION, USE AND EVALUATION
OF A MICROECONOMICS COMPUTER GAME

J. William Hanlon
Winona State College
Winona, Minnesota 55987
Telephone: (507) 457-2051

Donald P. Cole
Drew University
Madison, New Jersey 07940
Telephone: (211) 377-3000

The purposes of this paper are (1) to describe Micromod, a computer game designed as an aid to teaching intermediate microeconomics, (2) to describe some of the mechanics of using Micromod, and (3) to report the results of an empirical study designed to answer some questions about the validity of using micromod as a criterion for determining the course grade[1]. This question of validity is significant because grading provides the incentive for serious economic analysis of micromod as it is played, but if high performance in micromod were not related to competence in economics, one would certainly question the wisdom of using it as a basis for grading.

Introduction

Microeconomics is that division of economics generally concerned with the individual and collective behavior of business firms and consumers as they function to accomplish the allocation of the resulting economic goods and services among consumers. Of primary concern is the free market environment where the system of prices, free to respond to the actions of firms and consumers, provides the network of communications that integrates these actions to form a complete economic system. While microeconomics is ultimately concerned with the economy as a whole, the great bulk of discourse in the field is concerned with analysis of small segments of the economy as represented by theoretical models of firms and consumers set in alternative economic environments.

The traditional approach to teaching microeconomics at the undergraduate level is to study concepts and relationships within the context of theoretical models, while attempting to demonstrate relevance by pointing to examples taken from the real world. There are at least two major shortcomings to this approach. First, it is extremely difficult to locate real-world examples that clearly demonstrate basic economic concepts and relationships. This does not mean these concepts and relationships do not exist, but rather that they are obscured by many extraneous factors, both economic and non economic, so that any single economic concept or relationship cannot be isolated and observed. Indeed, the reason economists rely so heavily on models at all is that the real world itself is much too complex to analyze directly.

The second shortcoming of the traditional approach is that the dynamics of the models may receive little or no attention. Primary concern tends to be toward constructing models and describing equilibrium conditions, with little attention given to the behavior that causes a model to move toward its equilibrium state, or to the states it passes through enroute to equilibrium. Consideration of such dynamics would give the student a more complete and permanent understanding of economic models.

Micromod was designed to remedy these two shortcomings of the traditional approach. First, it provides a means whereby the student can directly observe the implications of a few basic economic relationships, just as the chemistry laboratory experiment enables the student to isolate and observe relationships between certain chemicals in a controlled and known environment. Second, the student supplies the behavior, given certain goals along with the environmental conditions specified in the computer program, that results in observable outcomes. Specific behavior is readily associated with specific outcomes. The student is a part of the processes of change and adjustment that occurs as the model moves toward its equilibrium state. This hopefully results in a better understanding of these processes, and of economic models in general.

Description of Micromod

Micromod consists of three separate games based on computer simulations of several microeconomic models. Each game, pure competition, pure monopoly, and oligopoly, represents a particular environment in which a product is produced and sold. Each environment is described by a single equation built into the computer program. The equations are:

Pure Competition: $P = A + B(S_i) + E(M_i) + FR + HY$

Pure Monopoly: $D_i = A + KP_i + WM_i + FR + HY$

Oligopoly: $D_i = A + B(S_i) + KP_i + E(M_i) + WM$

Where: P = Market price
A = The intercept
 $\sum S_i$ = Total amount of product supplied by all firms (students)
 $\sum M_i$ = Total amount spent on advertising by all firms
R = Price of a related product
Y = Per capita income
 D_i = Demand for the product of a particular firm (student)
 P_i = Price charged by a particular firm
 M_i = Advertising expenditures by a particular firm

All other symbols represent coefficients which can be changed from time to time in the computer program, and which values may or may not be revealed to the students.

A fine for violation of antitrust laws is levied in oligopoly whenever the demand for the product of any firm is larger than 30% of the total output of all firms. This adds an element of reality faced by real-world firms.

A set of five cost functions is included in each game. Each function includes three equations, one for total cost, a second for average total and a third for marginal cost. Each is of the form favored by the conventional microeconomic theory with zones of increasing and diminishing returns. Each function is identified by its fixed cost.

Mechanics of Using Micromod

Each student acts as a firm in each game, and is given a goal, such as to maximize profit or to maximize market share. The decision variables, manipulated by the student, in pure competition are the amount of product supplied and the level of advertising expenditures. The decision variables in pure monopoly and oligopoly are these two plus the price charged for the product. In addition, students must select which of the five cost functions to use for producing in each game by specifying the level of fixed cost in each round of play.

Each game is introduced as the corresponding chapters are covered in the text. Eventually all three games are played concurrently through the remainder of the course as other topics are covered.

Each student's decisions are punched on a computer card, one card for each game for each round of play, for batch processing on the computer. Each student receives a computer printout for each round prior to the next set of decisions. The printout shows all factors that are relevant to analyzing results. The students can easily relate past decisions to outcomes, and can use that information for the next round of play. A summary sheet for use by the instructor is printed out for each round of play.

The precise values of the demand coefficients may or may not be revealed to the students. The instructor might reveal only a probability distribution for each coefficient, while not revealing the precise value in the computer program, or he may give no hint of these values, thus forcing the student to do some analyses to get estimates of them. It is less frustrating to the student, but perhaps less realistic, when coefficients are revealed. Even when revealed, the game retains its intrigue from term to term because the values can be changed periodically, resulting in a new set of optimal decisions each time. Presumably, the faster the student reacts to these changes, the better he understands the underlying economic principles.

The instructor can invent ways to make changes in these coefficients meaningful and interesting to students. For example, in pure competition the instructor might announce that a new contract has been signed to ship so many units of product to Japan as a result of removal of tariffs, and that this has caused the demand curve to shift X number of units to the right. The student will then be expected to react appropriately to this change in his next round of decisions. The junior author sent notice of such events through the campus mails, with the unexpected result of making Micromod a topic of major interest in the campus residence halls. The senior author has received calls well into the evening from students asking about certain fine points. One student called at 11:30 one night to announce that he had just successfully written and debugged a computer program that would print out the results of all possible profit maximizing decisions in pure monopoly.

Students may be given any number of goals, including profit, return on investment in fixed cost, market share, sales dollar, return to advertising dollar, and others. The instructor may

assign one or more of goals and evaluate student performance accordingly. Students may be instructed to compete with each other, or they may be given some absolute standard for which to strive.

Empirical Study

The use of micromod as one of several bases for determining course grades provides incentive for the students to put forth a full effort in making his decisions. Without this incentive, experienced teachers know that the game would become a frivolous endeavor for most students, and very little economic analysis would be used for making decisions. Since this incentive is essential, teachers must question whether micromod performance is a valid basis for grading. The criterion used to judge validity in this study is whether micromod performance is an indicator of achievement in economic understanding. A second question to ask is whether performance in micromod is biased toward students with mathematics backgrounds.

Evidence relating to these two questions was obtained from analysis of data collected from two classes of the author at Drew University. A linear multiple regression equation was estimated for each class. The results are shown below:

Winona State College:

$$Y = 48.189 + 7.390X + 2.318Z$$

Coefficient of multiple determination = .250

Drew University:

$$Y = 64.800 + 6.260X - .440Z$$

Coefficient of multiple determination = .445

The variables are:

Y = An index measuring performance in the three micromod games

X = Average grade in the principles of economics courses

Z = Average grade in college mathematics (4.00 = A, zero if no college math had been taken)

The regression coefficients for X were significantly different from zero at the 5% level in both cases, while the coefficients for Z were not significantly different from zero. The Y variable, performance on micromod, is a measure of profit earned over several rounds of play in pure competition and pure monopoly, and for oligopoly it is a measure of the quality of a written report describing the economic analysis used for each decision. Oligopoly was graded this way because, even though a student may base his decisions on sound economic analyses, he may not obtain high profits because his outcomes depend largely on the behavior of other firms in the industry. These measures of performance were combined into a single index to obtain variable Y.

Two conclusions drawn from the analysis are.

1. There is a statistically significant positive relationship between micromod performance and general competence in economics.
2. There is no relationship between micromod performance and mathematical background.

The first conclusion suggests that performance in micromod seems to be an indicator of competence in economics, so it would seem appropriate to base part of the course grade on micromod performance. It may even be appropriate to dispense with the standard lectures on market structure and cost analysis and use micromod as a self-teaching device. This would free the instructor to concentrate on other more sophisticated topics.

The second conclusion suggests that the student of mathematics does not have a discernible advantage in micromod. This is important because the mathematics background of students in intermediate microeconomics in many schools includes the full range from little or no college training to several courses above the calculus. The non mathematics student appears to suffer no disadvantage in micromod, thus adding to the validity of using micromod performance as a basis for grading achievement in economics.

Summary

This paper includes a description of the structure and use of micromod, a computer game designed for the intermediate microeconomics course. Micromod consists of three separate games, based respectively on the market environment of pure competition, pure monopoly and oligopoly. Grading of game outcomes is essential to provide the incentive for serious student effort in the game, so instructors should ask whether micromod performance is a valid basis for grading achievement in the course. An analysis of empirical data, using linear multiple regression techniques concluded that micromod performance is a valid basis for determining course grades since it seems to be related to achievement in economic understanding, but is not related to mathematical background.

FOOTNOTE

1. This paper represents part of the on-going project of designing and evaluating micromod. Assistance in this project has been obtained from the General Research Fund of Winona State College and from the Joint Council on Economic Education. Adolph C. Nydegger, College of St. Teresa, and Paulette A. Cichon, formerly of Northern Illinois University, wrote the computer programs for Micromod. Dr. Hanlon designed the game and has used it in several classes during the past two years.

NEW APPROACHES TO BUSINESS SIMULATIONS

JAMES D. J. HOLMES
St. Andrews Presbyterian College
Laurinburg, North Carolina
Telephone: (919) 276-3652

New Approaches

In looking at new approaches that are being taken in simulation the first illustration that comes to mind is intercollegiate competition. After a slow start intercollegiate competition is beginning to spread into the hinterlands rather than being the tool of a few large universities. Two of the oldest such competitions are those run at Michigan State University and Emory University. The Michigan State game covers extended play for a full semester while Emory restricts its competition to six weeks. In both cases decisions are filed by mail or TWX with a wrap-up conference at the end of play featuring awards to the top teams.

In North Carolina we have gone an additional step in establishing several intrastate competitions. Under the sponsorship of the North Carolina Educational Computer Service (NCECS) a two day game was run in October, 1970. Primarily for faculty members from colleges and universities throughout North Carolina response was overwhelming and provided the impetus for the first student competition which was held in February, 1971. At the student conference there were 27 schools represented by 3-member teams. The Executive Game[1] (TEIG) by Heushaw and Jackson was chosen for both faculty and student conferences because of its comprehensive nature and yet relative ease of play.

Cash prizes were awarded to the three top student teams by a North Carolina bank and an award of a small amount of free CPU time was given by NCECS to the institution sponsoring the top team. It would be an understatement to say that the response to this competition was overwhelming. All decisions were made on the premises of the Triangle Universities Computation Center over a span of two days. Excitement was at a high level during the entire session and peaked quite high at award time on the afternoon of the second day.

Based on the success of this first student competition plans are being made at the time of this writing for a continuation of the intrastate competition. During the spring of 1972 a competition based on TEIG and played on site over two days will be held for junior and community colleges in North Carolina. These institutions are quite enthusiastic over the use of simulations but their students are at a disadvantage when competing against students from four-year colleges and universities. Later, during the fall of 1972, a game and conference using INTOP-International Operations Simulation[2] by Thorelli and Graves will be staged for four-year colleges and universities. Play will extend over much of the fall semester with decisions being submitted through terminals tied to the Triangle Universities Computation Center. The game will then culminate with a one-day conference for awarding of prizes and critique of game play. It is anticipated that participation in these two competitions will be quite extensive and spirited.

In administering different simulations, it has been my observation that more often than not students sub-optimize their decisions because competition is not keen enough to force them to do otherwise. In 1963 Thomas Hoffmann of the University of Minnesota began work on the concept of using the computer as a participant thus causing students to do a better job of decision-making. Using the analogy of par in golf, Hoffmann established a system of heuristic rules which gave the computer player superiority not because of greater knowledge of the mathematical structure of the game but rather "as a result of its inherent requirements of consistency, comprehensive analysis, and logically derived decisions." The vehicle used for this experiment was the IBM Management Decision-Making Laboratory, Model 1. Professor Hoffmann is not presently engaged in research of this nature but the original game package has served as the basis for three recent doctoral dissertations.[3]

After the time I learned of Hoffmann's work I had begun to brainstorm with a couple of students over the idea of letting the computer use regression analysis to become a participant. Learning of Hoffmann's research strengthened my belief and during the winter of 1971 I began writing a subroutine for an existing simulation which would allow the computer to become a competitive participant. My hypothesis was that the computer can generally do a better job of decision-making than most students if it has the possibility of using objective mathematical tools which most students will not be using, and that students are basically competitive and will do a better job when facing dramatic defeat by a machine. In the initial part of the project a subroutine was written to incorporate regression analysis into The Executive Game. Heuristic rules were then established which would use the results of regression in making decisions.

In actual play decisions relative to price, marketing expenditure, and research and development expenditure are developed as variants from the mean of each of these factors for all of the teams. Demand is then estimated by the equation

$$Y = a + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4$$

with b_1, b_2, b_3, b_4 being correlation coefficients relating the independent variables of price, marketing, research and development, and economic index to the dependent variable of demand. Other decisions such as production scheduled, raw material purchases, and maintenance expenditures logically follow. Since regression analysis requires observations of historical data the computer buys such data with other teams being given the same opportunity.

The first run of this package, referred to as TEXGCT (the Executive Game - Computer Team), was made in the summer of 1971. During the previous spring a senior seminar had participated in TEXG with eight teams for twelve periods of play. These decisions were replayed in the summer against the computer team. Such an approach left no chance for interaction or learning on the part of the non-computer teams but did provide a valuable first test of the hypothesis. TEXG incorporates a year-end run in the regular game which evaluates teams on the basis of profit earned and dividends paid assigning a rate of return to each team. The computer team placed second out of nine teams with a return of 2.219%. The leading student team showed a return of 2.234% and the third place student team showed a return of 1.135%. The other student teams fell off sharply with two teams showing negative returns.

At the beginning of the fall semester, 1971, eight volunteer student teams were recruited and an interactive competition began. Since the teams were made up of volunteers there was no worry about grades or any faculty pressure. During the twelve periods of play two teams were forced into absolute bankruptcy. A student team placed first with a return of 3.419%. In second place the computer team showed a return of 2.226% and the third place student team showed a return of 2.098%. It is necessary to mention that bias exists in these results in that one member of both the first and third place teams had participated in the spring play. The rate of return for other teams fell off sharply with one team at 1.383% and the other three below 1.0%.

During the spring of 1972 TEXGCT will be used as a partial requirement for the senior seminar. It will be interesting to observe this competition wherein the students will feel the pressure of grades, their peers, and the computer team, all at one time. Hopefully, such an experience will give them a sampling of pressures that exist in the business world.

Too often in business simulations authors and publishers have tended to overlook the potential afforded by role-playing while emphasizing the decision-making process. Some authors pay lip service to the need for role-playing by including with their game the request that a president, vice-president, and secretary-treasurer should be decided upon early in the game and that the president should be responsible for having the decision form completed when the administrator comes for it. The best game play I've ever administered was a situation in which the decision-making portion of the game was somewhat weak but gave the opportunity for labor-management confrontations and negotiations. With two hard-boiled faculty members as union representatives, the negotiations often became too realistic to be enjoyable from an administrator's point of view. After participating in that particular simulation one student recommended that I add the roles of supplier of raw materials and consumer of finished goods to those of union and management with the supplier and consumer being external to the game. The student further proposed that participants would shift functions during the game with each participant getting to play each role one-fourth of the time. Time has not permitted me to explore the proposal but I think there is considerable merit in such an idea.

In using simulation as a teaching device I have been faced with the problem of grading participants. For many years I rejected every scheme that I could think up after a one-time trial. During 1970 I moved to peer grading and was well satisfied after using it during one course. The same approach was also used during the spring semester of 1971 and will be used during the spring of 1972.

The sample form at the end of this paper shows the approach I have taken. In summary, each student on a team provides a quantitative valuation of himself and his team members based upon quality and quantity of contribution to team effort, accuracy of work performed, dependability, and initiative. Each student is then further asked to provide a prose comment on overall performance for each of his team members. At this point I enter the grading process through valuation of the prose comments. My effect is small, however, due to a weight of only one out of a total of eleven being assigned to my rating of the prose evaluation. For the total simulation I then use the following weights:

Individual Peer Rating	40%
Team Rank in Final Results Based on Profit Produced	30%

DATE: _____ PHASE: _____

SCALE OF SCORES

<u>FACTORS</u>	<u>WEIGHT</u>	<u>MEMBERS' POINTS</u> (MULTIPLY SCORE TIMES WEIGHT)				
		<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>
1. Quality of Contribution	3	_____	_____	_____	_____	_____
2. Quantity of Contribution	3	_____	_____	_____	_____	_____
3. Accuracy of Work.....	2	_____	_____	_____	_____	_____
4. Dependability.....	1	_____	_____	_____	_____	_____
5. Initiative..... Sub-Total Points	1	_____	_____	_____	_____	_____
6. Quality of Rating.....	1	_____	_____	_____	_____	_____
<u>TOTAL POINTS</u>		_____	_____	_____	_____	_____
<u>RANK</u>		_____	_____	_____	_____	_____

<u>NAME</u>	<u>PROSE COMMENT ON OVERALL PERFORMANCE</u>
MEMBER A: _____ (FILL IN NAME)	_____ _____ _____
MEMBER B: _____ (FILL IN NAME)	_____ _____ _____
MEMBER C: _____ (FILL IN NAME)	_____ _____ _____
MEMBER D: _____ (FILL IN NAME)	_____ _____ _____
MEMBER E: _____ (FILL IN NAME)	_____ _____ _____

FIGURE 1. Peer Rating System.

Team Presentation of Strategy and Results Before Panel of Judges 20%

Team Evaluation by Instructor 10%
100%

Random Thoughts for the Future

I do not profess to have special insight into the problem areas of simulation; however, there are several areas that obviously need development or improvement. One of such problems is the general unwillingness of publishers to accept any responsibility for providing information other than what is furnished in the student and administrator manuals. Some games are well documented as to optional features and requirements for implementation on computer systems other than the one for which the simulation was originally written. However, other well publicized games are very poorly documented. In one instance I talked with the "author" of a game who after a few minutes of questioning admitted he knew little about the program but suggested I call Joe Smith who did the programming four years ago and was now reported to be in Houston, Texas! I feel that it is mandatory that publishers acquaint themselves well enough with simulation techniques and documentation requirements to establish standards to be set by any author seeking publication of a game.

Relative to the actual running of a game, I feel that we users are generally guilty of poor administration. More enjoyment and improved learning result from an administrator who is not only enthusiastic about gaming but is well versed in the details of the game being administered.

A final area which must be developed before simulation can take its rightful place as a powerful teaching tool is the dissemination of user information on a state, regional, and national basis. We badly need to know what is being done in research leading to development of new and different simulations. We further need to know both objectively and subjectively the experiences of users of published games. If certain games are poorly documented and hard to implement, this should be conveyed to the publisher and to our colleagues. Obviously we must use discretion and tact with a purpose of producing better simulations rather than blasting a particular publisher or author. It is my belief that a free exchange of information can tremendously increase the quantity and quality of simulations and tremendously increase the number of users of this very important tool.

REFERENCES

1. Henshaw, Richard C., Jr., and Jackson, James R., The Executive Game. Homewood, Illinois. Richard D. Irwin, Inc., 1966.
2. Thorelli, Haas B. and Graves, Robert L., INTOP-International Operations Simulation. New York. Free Press of Glencoe, 1964.
3. Braasch, John, "Business Games, Programmed Player, and Individual Decision-Making Profiles," University of Minnesota, 1966.
4. Eliason, Alan, "A Study of the Effect of Formal Training in the Application of Quantitative Models to Production as Revealed by an Interactive Decision Simulation," University of Minnesota, 1970.
5. Schmid, H. Giles, "An Experiment on the Reliability and Validity of the Performance Measure, Profits, in Two Management Games" University of Minnesota, 1969.

COMPUTER SIMULATION OF ECONOMIC MODELS FOR INSTRUCTIONAL USAGE

Stanley Wilson and Ray Billingsley
Texas A&M University
College Station, Texas 77843
Telephone: (713) 845-5443

Introduction

An academic discipline is a system of concepts and hypotheses. These concepts and hypotheses are viewed separately as abstractions from reality and constitute, for the most part, the subject matter of a discipline. Most disciplines view this part of their subject matter as theory. Other concepts consist of the relationship between these abstractions. This latter is particularly crucial to a discipline because it contains the discipline's way of approaching reality, i.e., its technique of analysis. The facts of reality and thus the abstractions from reality change, but, presumably, the techniques of analysis continue to be valid. They may be added to and improved but hopefully not invalidated. To the extent that an individual knows these methodological concepts - to that extent he is a practitioner of the field. It follows that training the student in a field consists of imparting these methodological concepts to him and giving him practice in using them.

In order to teach methodology there must be a transportation (i.e., communication) of these concepts from the intellect of the instructor to the intellect of the student. Direct communication of the intellect of one individual to the intellect of another is (at least in our current state of technology) impossible. Thus, the concepts must be embodied in some physical medium and this medium presented to the student's senses in the hope that he will abstract the concept from the medium. It follows that the medium must be chosen and used with the objective of maximizing the ability of the student to abstract from it. Each particular medium or language has particular advantages and disadvantages. The optimum choice may well be several media used in combination.

This analysis applies at least as well to economics as to other disciplines. Economics, in particular, makes use of several languages in combination to convey its concepts and techniques of analysis. These include the written or spoken word, graphs and mathematical formulas. The concepts of economics relate various elements in the economy in a cause-effect relationship. For example, the concept "demand" relates price (the cause) with quantity purchased (the effect). Demand may be conveyed in words, in a mathematical formula or in the form of a graph.

The basic methodology of economics consists of building and using economic models. An economic model may be defined as a system of economic concepts. The model purports to be an abstract representation of a part or all of an economic system. The technique of economic analysis consists of four steps: (1) concepts are defined, (2) concepts are put together (or related) to form models, (3) changes are introduced in the models, and (4) the effects of these changes are noted. Economic models serve two purposes. First, they are a method by which economists can perform experiments. Unlike the physical scientist, social scientists cannot experiment directly on that area of reality they are studying. Secondly, economic models give students practice in using economic concepts (i.e., practice in performing economic analysis).

A computer model is a particularly effectively medium for allowing students to introduce changes and observe the effects. This is because a computer model (or program) is a sequence of instructions to the computer telling it to simulate or act like the economic entity being modeled. If the computer is to accurately conform to the set of concepts in the intellect of the economist, then it is vital that consideration be given to how these concepts can be embodied in a computer program. To illustrate the problems of programming economic concepts, several concepts and approaches to programming them will be examined.

Market Equilibrium

The first and simplest model presented to economics students in the simple market model. This consists of three concepts: (1) demand, (2) supply, and (3) equilibrium price and quantity. The concepts, demand and supply, are easily embodied in a program because they are already in the form of mathematical equations.

$$\text{Quantity Demanded} = f(\text{Price})$$

$$\text{Quantity Supplied} = f(\text{Price})$$

These become:

$$QD = AD - BD \times PED$$

and

$$QS = AS - BS \times PES$$

where the parameters for demand are AD, BD and ED and those for supply are AS, BS and ES. Other forms of these equations may be utilized also. Allen gives several forms of the demand curve (2, p. 114). There exist several computer languages (including FORTRAN) into which it is easy to translate such functional forms.

The concept of market equilibrium or movement toward equilibrium is another matter, however. There is no mathematical statement of this process. The condition of equilibrium is easy to represent. It is that quantity demanded equal quantity supplied ($QD = QS$). But, the real process of how equilibrium is actually reached remains a mystery. Walrus asserted a tatannement or "groping" toward equilibrium but no one seems to have gone beyond this, including modern economists. See Patinkin, for example, (10, pp. 38-43).

One can, however, deduce several characteristics which the equilibrium process must have if it is to actually achieve or arrive at that price where $QD = QS$. First of all, the equilibrium process must essentially be a process of changing the price. Second, the direction of the price change must depend on which is larger, QD or QS. If $QD > QS$, the price must go up. If the reverse is true, the price must decline. Finally, the size of the change in price must be proportional to the size of the surplus or shortage. Letting DELP be the change in price, the following equation achieves these conditions.

$$DELP = \frac{QD - QS}{(QD + QS) / 2} \times KDP$$

All terms in this equation are positive so that if $QS > QD$, the $(QD - QS)$ will be negative and thus DELP will be negative (the change in price will be declining). Notice that the value of DELP depends not on the absolute size of $QD - QS$ (the surplus or shortage) but on the relative size. One intuitively feels that the power of the force making for the price change depends not on the absolute size of the surplus but on the size of the surplus relative to the total size of the market. The size of the market is interpreted to be the amount bought and sold (i.e., QD and QS). The size of the market is thus interpreted to be $(QD + QS)/2$; i.e., the average of QD and QS.

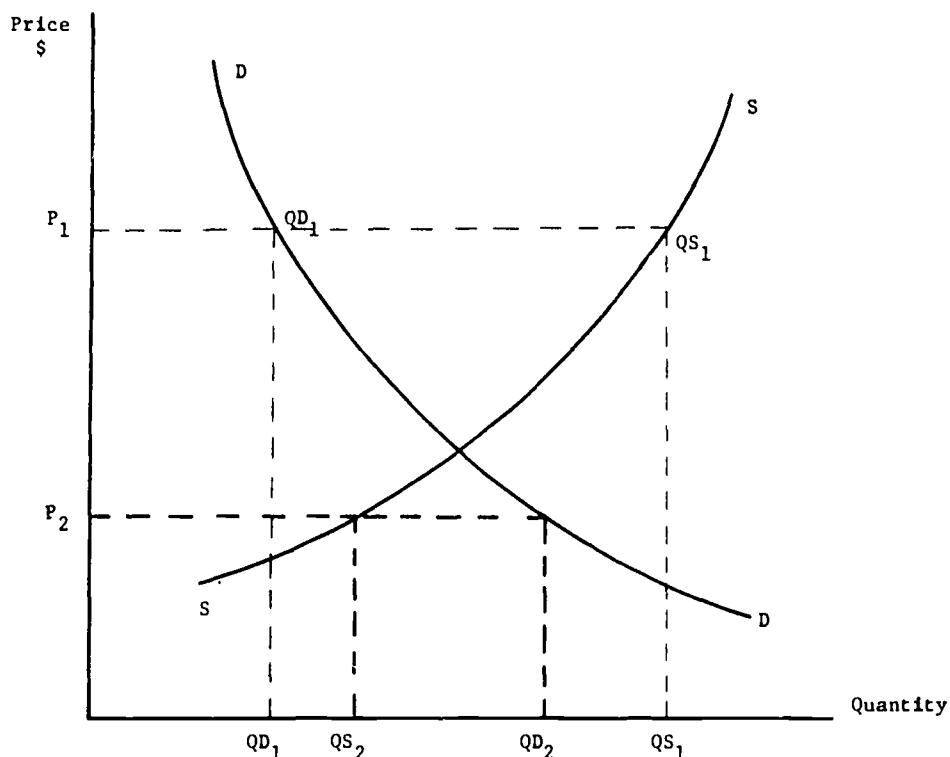
An effect of this formulation is that as the price approaches the equilibrium price, the size of $(QD - QS)$ becomes smaller, relative to $(QD + QS)$, thus DELP becomes smaller and the price converges on equilibrium. In this way the price doesn't oscillate about the equilibrium indefinitely. KDP is the constant of proportionality. It may be set to 1.0 for a normal market. For a cobweb effect KDP can be made larger. The larger the value of KDP, the greater the oscillation. Thus, the model can be made to converge on the equilibrium price or oscillate as one wishes. Even if KDP had a value of 1.0, a cobweb effect could be achieved if one selected functional forms and/or parameter values which produced the appropriate elasticities of demand and supply.

A program (called MARKET) has been written, utilizing the formula discussed above, and has been used to teach the introductory principles course. It seems to have been successful in demonstrating the equilibrium process. Each student chooses a set of parameters for the demand and supply functions as well as an initial price. The initial price is used to calculate QD and QS. These are then printed out along with the price. The program then calculates DELP and adds it to P. Then it uses this value to recalculate QD and QS and print them out along with the new price. The last few lines of printout show no appreciable difference in QD and QS and thus no appreciable difference in price. This demonstrates to the student the concept of equilibrium. That is, if the price is not at equilibrium, it will converge on equilibrium and when it reaches equilibrium, it will remain there. This demonstration is particularly effective because the student gets to select parameter values and the initial price. This demonstrates that the system will seek equilibrium for a wide variety of initial conditions.

There is no implication that the equation given above "explains" how the market reaches equilibrium. But the equation does make the model perform like economists envision that the market performs. If the price is to converge on equilibrium, then the changes in price must be smaller as price gets closer to equilibrium. One way to achieve this is to make the change in price proportionate to the difference between the price and the equilibrium price. This could be done by solving for the equilibrium price but it would imply that the market "knew" the equilibrium price. The function outlined above assumes only that the price change is influenced by the relative size of the surplus or shortage.

Shifts in Demand and Supply

MARKET also allows the student to introduce changes in demand and/or supply. This is accomplished by introducing changes in the parameters of the functions. The model first seeks the equilibrium price and then, once this is achieved, the change in the



At Price P_2 : $QD_2 > QS_2$

Thus $(QD_2 - QS_2) > 0.0$

and $DEL P > 0.0$

therefore P will increase

At Price P_1 : $QS_1 > QD_1$

thus $(QD_1 - QS_1) < 0.0$

and $DEL P < 0.0$

therefore P will decline

function parameter is introduced. If the change is in the demand function, then, at the equilibrium price achieved, quantity demanded is no longer what it was. Thus, $QD - QS \neq 0.0$ and the program prints QD, QS, and price, changes price again and seeks the new equilibrium. If the change is in a supply function parameter, a similar process ensues. In this way, the student can introduce changes in the model and observe the effect on the model.

The different parameters have different effects on the functions. AD and AS change the price intercept of their functions and shift them in a parallel manner. BD and BS change the slope or angle of their functions. ED and ES change the curvature of their functions. Together, these six parameters give the student the ability to change the demand and supply in any conceivable manner.

A second program, SIMPLE FIRM, simulates a firm with cost and revenue function. It makes use of some of the concepts described for MARKET. The firm's average revenue function is:

$$AR = A - B \times Q^E$$

where AR is average revenue, Q is the quantity of output and A, B and E are parameters. The marginal revenue is the first derivative of total revenue. Average variable cost is:

$$AVC = XQ^2 - XQ^2 + Z.$$

Marginal cost is the first derivative of total cost. The student specified a beginning quantity of output as well as the parameters for the cost and revenue functions. If $MC \neq MR$, then the quantity is changed by DELQ. The formula for DELQ is:

$$DELQ = \frac{MR - MC}{(MR + MC) \times 0.5} \times CDQ.$$

The above formula has the same form as that for DELP in MARKET and for analogical reasons. The function compels DELQ to converge on the profit maximizing level of output (when MC = MR) by smaller and smaller amounts. This implies that if the firm is at an output level where MR > MC, it will expand output and that the change in output will be proportionate to the difference between MC and MR. A similar implication holds when MR < MC. The program also calculates total revenue, total cost and profit. This is particularly useful because the student can observe that profit increases as the program moves closer to the quantity where MC = MR. One useful approach is to rerun the program with the same set of parameters but with the initial output level being on the other side of the profit maximizing level of output from the first initial output level. For example, if the profit maximizing level of output was 120 units and the first initial output level was 150 units, it is useful to rerun the program with an initial output level of about 90 units. In this way the student sees that profit increases as the output level approaches 120 units no matter from which direction and thus that profit is maximized where MC = MR. This has proved to be a useful way to reinforce this important, and by no means easy to grasp, concept. Indeed, SIMPLE FIRM has proved to be substantially superior to written or graphic demonstrations and explanations of this concept.

A third program, FACTOR-PRODUCT, simulates a firm's economic responses associated with a production function of the form

$$Y = f(X).$$

The program lists the parameter values for the production function and the prices assumed for factor, X and product, Y. Starting with a beginning value for X, the program calculates and prints all of the relevant economic variables for each iteration as the calculations move toward equilibrium. These include: (1) the quantity of X, QX; (2) the price of X, PX; (3) total fixed cost, TFC; (4) the quantity of Y, QY; (5) the price of Y, PY; (6) total revenue product, TRP; (7) the net revenue, NR; (8) the marginal physical product, MP; (9) the value of the marginal product, VMP; (10) the average product, AP; (11) the average revenue product, ARP; (12) the marginal cost, MC and finally (13) the stage of production the calculation refers to. These stages are indicated as 1, for increasing returns to scale, 2, for diminishing returns to scale and 3, for decreasing returns. An option available on the data cards allows the program to run past the point of economic optimum so that all stages of production can be illustrated if the production function used allows for all stages. In its present form, four types of production functions are included in the program. These are:

- (1) Cobb-Douglass

$$Y = Z \times X^B$$

- (2) Quadratic

$$Y = A + (B \times X) - (C \times X^2)$$

- (3) Quadratic Root

$$Y = A + (B \times X^{0.5}) - (C \times X)$$

- (4) Cubic

$$Y = A + (B \times X) + (C \times X^2) - (D \times X^3).$$

FACTOR-PRODUCT searches for that level of usage of the resource X, such that profit is maximized. This position is achieved when two conditions are met. They are (1) PX = VMP (in the case of pure competition) or MC = VMP and (2) VMP is declining. This is a particularly interesting condition because with many production functions a VMP curve will be generated such that PX = VMP at two positions. When FACTOR-PRODUCT locates such a position, it checks to see if VMP is declining about that position as X increases. If it is not, it continues to search by increasing X. This is an example of embodying economic principles into a computer program.

FACTOR-PRODUCT, like all of the other simulations, must have some rule for deciding how much to change X. The formula is similar to the ones used before.

$$DELX = \frac{MC - VMP}{(MC + VMP) / 2} \times CDQ$$

The output demonstrates to the student that as X gets closer to the level where VMP = MC, the profit rises to a peak.

A Macroeconomic Model

Another model written and utilized was a simulation of a simple Keynesian macroeconomic model. This program, called KEYNES, besides having a consumption function also has a marginal efficiency of capital function and an interest rate determined by a money market. The money market is of particular interest because it has a liquidity trap. See Ackley (1, pp. 192-194, 384-385).

The user specifies a beginning income level and interest rate as well as parameters for the consumption function, MEC curve, liquidity demand for money curve and transaction demand for money, taxes and government expenditures. The program calculates the consumption and investment levels and sums them along with government expenditures to get total expenditures. Investment is determined by the interest rate through the MEC curve. The interest rate chosen is the equilibrium interest rate which is determined by the money market. The money market functions in a similar manner to the process described for MARKET above. The only exceptions are that the supply of money is perfectly inelastic (and given by the user) and that below a certain interest rate (specified by the user as an option) the demand for money is perfectly elastic.

Once total expenditures are calculated and the other values calculated, they are printed out. Then the program sets income in the next iteration equal to total expenditures, and the values are recalculated. This continues until equilibrium is reached (where total expenditures equal income). The path toward equilibrium is not necessarily a straight one. The changes in income not only change consumption but also the transactions demand for money. This shifts the total demand for money, thus changing the equilibrium interest rate and investment. Thus, a convergence toward equilibrium may follow a cyclical path.

Once the model reaches equilibrium, it introduces any changes the user specified in function parameters or amount of taxes or government expenditures. Then it does the calculation again and begins a movement to the new equilibrium position. The movement toward equilibrium income does not require the calculation of a change in income such as MARKET requires the calculation of a change in price. The new income level is determined by the previous expenditure level. The rule that expenditures in one period equal income in the next period automatically assures a movement toward equilibrium income. This feature assists in conveying that important principle of macroeconomics. Unlike MARKET and SIMPLE FIRM, KEYNES is not suitable for an introductory course, but is suitable for an intermediate macroeconomics course. Like the other two models, KEYNES has been used in economics classes.

Classroom Usage and Testing

No matter how well a computer model simulates a part of the economy, if it is to be a useful teaching technique, it must be integrated into the course. That is, the model must be an integral part of lectures, discussions, outside class activities and testing.

To do this one begins by emphasizing in his lectures that economic concepts and models can be communicated in several languages, that each language has its advantages and disadvantages and that computer codes are one medium or language in which economic models can be expressed. The course becomes a sequence of models. The lecturer begins each model by defining the concepts in several different languages (words, graphs and equations). Then he puts the concepts together to form models in these various media, introduces changes in the models and observes effects. This gives a general outline to the lectures, i.e., the series of lectures on each model can be divided into the four steps referred to earlier, with computer output used as an exhibit when one comes to the lectures on introducing changes and observing effects.

Students can be assigned the task of selecting a set of original model parameters and one or more changes in parameters. These can then be run using the program deck and the results used by students as the basis of classroom discussion or as an extra example to be used, along with their notes and the text, for studying. A second subject of discussion is the flowchart. The instructor can treat the flowchart as another language or medium of expressing the model. Alternately, one can suggest that the class attempt to flowchart the model during one class period. This exercise is particularly useful for bringing out any weaknesses in the students' understanding of the model. Those who have programmed know that unless one knows the process or system precisely one cannot program it. Attempts to program, particularly at the flowchart stage, bring out any hidden defects in ones understanding. A person may believe he understands something until he attempts to explain it to a computer. The student can participate, at least vicariously, in this discovery of ambiguity by participating in the flowcharting process. Works on flowcharting are available. See, for example, Chapin [5] and Farina [6].

Computer models can also be used as part of lab sessions or as individual work. To make programs available and useful there are two requirements. First, the computer code must be available to the student. The code may be in the form of a card deck, or the program may be on the system. In either case, the emphasis must be on convenience of usage. The second requirement

is the availability of a good (i.e., clear and complete) documentation for each model used. There are standards of documentation which are available[4]. Each of the programs mentioned has been documented (with the exception of FACTOR-PRODUCT and its documentation is now being worked on) and references to these can be found in the bibliography.

It is unfortunate, but true, that most students are oriented toward tests. If something is covered in class but will not be on the exam, they give it only passing notice; allocating their time to those things they will be tested on. Thus, if computer models are to be used effectively in a course, they must somehow be on the examinations. One way this was done was to have a takehome exam in which the questions consisted of a computer printout. The examination process consisted of the following:

1. Each student chose numerical parameters for the model and also chose changes.
2. These were run on the computer and the printout for each student's parameters were given to him.
3. The student took the printout home and used it as the basis of his "examination."
4. He was graded on how well he explained the model and how well he explained what the model was doing (what his printout said).

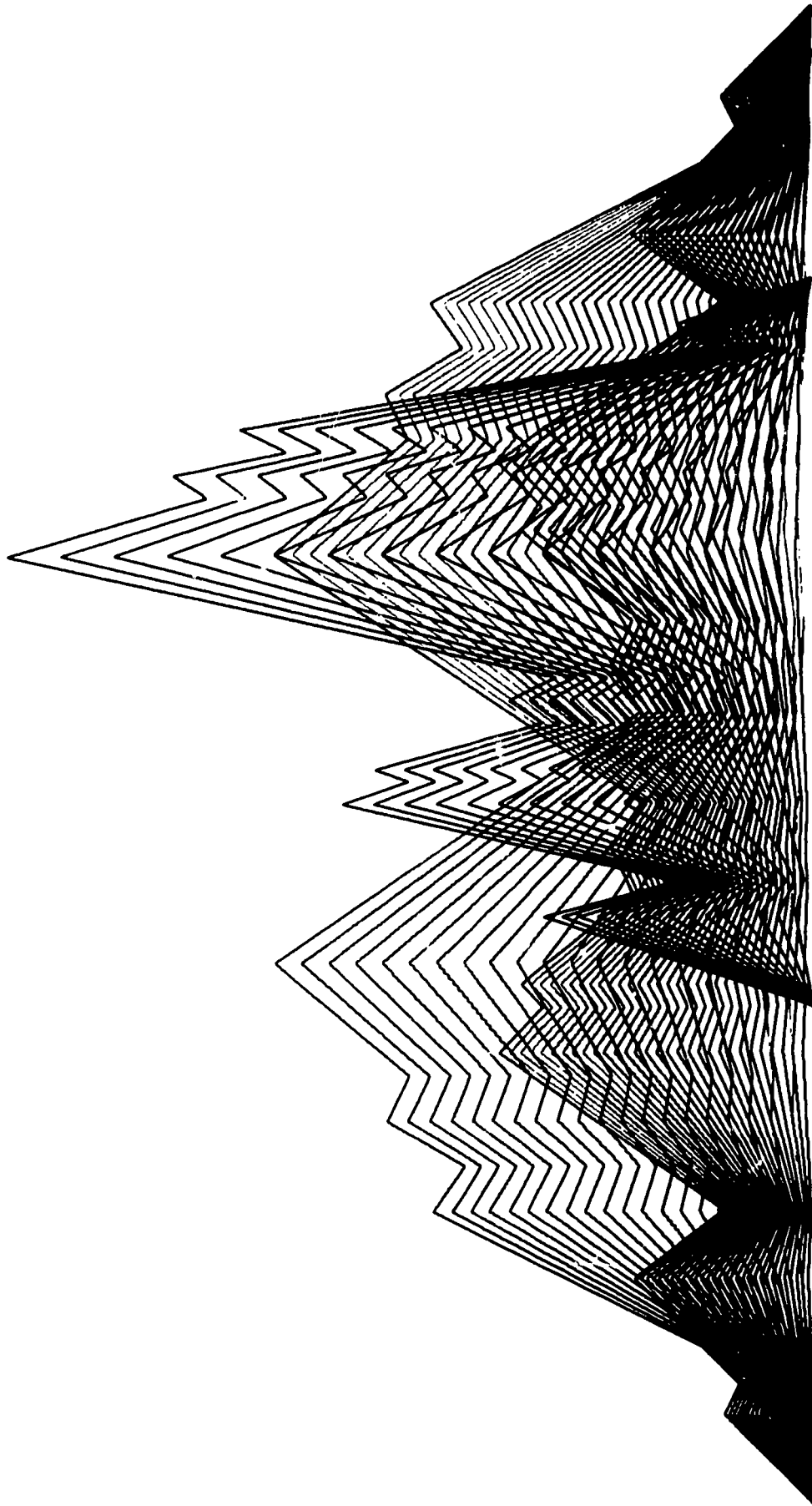
It should be observed that the students were not only having a takehome exam, they were also, in effect, writing their own exam questions. If a student caused his model to perform a number of complicated, interesting changes, then he had a number of complicated, interesting changes to explain, his work was of a higher quality because of the nature of the questions, and thus he got a better grade. Those who performed simple experiments had a less sophisticated explanation and thus received an average grade. There were also, of course, those who failed or received a 'D'.

This approach to lectures and examinations was tried out at Texas Southern University. The general effect seemed to be that students' grades improved by about one letter. That is, those who had been making C's now were making B's. However, not all students improved. It is difficult to determine whether the improvement was due exclusively to the use of computer models or whether the takehome exams afforded them the opportunity to express themselves better because they were not under tension. It is obvious, however, that the performance of the class was not lowered, and this was at the same time that they were receiving a more rigorous introduction to economics, an introduction to the use of computers and an introduction to computer modeling of economic systems.

REFERENCES

1. Ackley, Gardner, Macroeconomic Theory, New York: Macmillian Company, 1961.
2. Allen, R. G. D., Mathematical Economics, London: Macmillian Company, Ltd., 1965.
3. Bilas, Richard A., Microeconomic Theory: A Graphic Analysis, New York: McGraw-Hill, 1967.
4. Billingsley, R. V., and Stanley Wilson, Program and Model Documentation Standards, Program and Model Documentation 71-1, Department of Agricultural Economics, Texas A&M University, February 1971.
5. Chapin, Ned, Flowcharts, New York: Auerbach Publishers, 1971.
6. Farina, Maria V., Flowcharting, Englewood Cliffs, New Jersey: Prentice-Hall, 1970.
7. Ferguson, C. E., and Charles Maurice, Economic Analysis, New York: Richard D. Irwin, Inc., 1970.
8. Naylor, Thomas, Joseph Balintfy, Donald Bardick, and Kong Chu, Computer Simulation Techniques, New York: John Wiley and Sons, Inc., 1966.
9. Orcutt, Guy H., "Simulation of Economic Systems," American Economic Review 59 (Dec, 1960), 893-907.
10. Patinkin, Don, Money, Interest and Prices, New York: Harper and Row, 1956.
11. Samuelson, P. A., Foundations of Economic Analysis, Cambridge: Harvard University Press, 1948.

12. Weintraub, Sidney, Price Theory New York: Pitman Publishing Corporation, 1949.
13. Wilde, D. J., Optimum Seeking Methods, New Jersey: Prentice-Hall, 1964.
14. Wilson, Stanley E., Economic Model - Market, Computer Program Documentation for Texas Regional Academic Computer Experiment, Department of Industrial Engineering, Texas A&M University, Copes Form No. 13.
15. Wilson, Stanley E., Economic Model - Simple Firm, Computer Program Documentation for Texas Regional Academic Computer Experiment, Department of Industrial Engineering, Texas A&M University, Copes Form No. 14.
16. Wilson, Stanley E., and Stedman Cary, Keynes-Computer Simulation of a Macroeconomic Model, Computer Program Documentation for Texas Regional Academic Computer Experiment, Department of Industrial Engineering, Texas A&M University, Copes Form No. 10.



W. L. Bouttote

MOUNTAINS by William (Bud) Bouttote
Problem: Continuous line design/non-maze pattern

COMPUTER ASSISTED INSTRUCTION IN ECONOMICS
AT THE UNIVERSITY OF NOTRE DAME

Frank J. Bonello and William I. Davisson
University of Notre Dame
Notre Dame, Indiana 46556
Telephone: (219) 283-6335

Introduction

The development of the computer represents one of the major technological changes of the last twenty years. Although colleges have had access to these machines since the very beginning of their development, only recently have attempts been made to utilize computers as learning tools. Indeed up until the last few years only selected students, namely physical science and engineering students, came into direct contact with the computer. The object of this contact was primarily to give the student an ability to utilize the services of a high speed calculating machine or electronic slide rule. Recently however, attempts have been made to broaden student use of the computer so that business, social science, and humanities majors are also exposed to the computer. More importantly attempts have been made to enlarge the role played by the computer within the educational process, to make it a viable learning tool[1].

These developments owe their origin to several different technical and educational advances including: an increase in faculty knowledge and appreciation of the computer, the development of simple computer languages, and the development and extension of interactive or time sharing systems. These last two elements represent major forces leading to change. The development of simple computer languages has reduced significantly the time which both students and faculty need to expend in order to acquire an ability to communicate with the computer. The development and extension of time sharing and remote terminal facilities has reduced wait time for answers almost to zero; indeed in some cases instantaneous two way communication between the computer and the faculty or student user is now possible. This of course has created many new possibilities for the use of the computer as an educational tool.

Within the context of the social sciences, a computer can function as a demonstration facility, as an automatic corrector with built in teaching assistance, and as a "real world" environment for the testing of alternative hypotheses. At most colleges there has been some implementation of computer assisted instruction (CAI) in each of these three areas. Unfortunately, this has generally occurred only in piecemeal fashion. To fully exploit the potential of CAI, it is necessary to establish a fully integrated system, a system that employs the computer in each of the three noted functional areas. In the Economics Department at the University of Notre Dame we are using such a system. The purpose of this paper is to give a report on our progress to date. More specifically we discuss each of the functional areas, indicate what we feel are pedagogical advantages and disadvantages, and provide examples drawn from elementary and intermediate level macro and micro economic courses.

The Computer as a Demonstration Facility

Frequently in Economics it is useful to demonstrate or underscore theoretical properties or conclusions by means of numerical examples. For instance in macroeconomics certain models are presented which have the property that a dollar's increase in government expenditure has more of an expansive impact on national income than a dollar's decrease in taxes. A teacher having presented such a model, indicated the particular property, and established the conceptual basis for such a conclusion might seek to reinforce this conclusion by a series of numerical examples. To do so during class time might involve too heavy a time cost and to assign the examples as homework involves the same heavy time cost and a tedium that both students and teacher wish to avoid. If however, the student has access to a computer terminal and is able to call a routine that will make the necessary calculations, the series of examples can be processed in less time than would be required to complete a single example manually. The instructor must provide the necessary input data. The student uses the computer to make calculations. The calculations are made and the answer given in such a way as to demonstrate a particular property.

In order to utilize a computer effectively in this way several prerequisites must be satisfied. First, the teacher must present the argument conceptually. In terms of the above example the instructor must not only present the appropriate model but also argue or explain the conceptual basis for the particular conclusion. Secondly, the instructor must assign problems and input data which obviously employ the same basic model and which numerically highlight the particular conclusion. Third, the computer routine utilized by the student must employ the same basic model and yield output in a form which underscores the particular conclusion.

The advantage of using a computer as a demonstration facility is that a large number of numerical examples may be processed in a very short period of time. Not only can a particular conclusion be demonstrated but the reasons why a particular result obtains can also be illustrated. This is accomplished by having the student knowingly switch models. Thus all changes within a model can be explored and compared with all changes within some other model. Time no longer imposes itself as a constraint and the tedium involved in making similar numerical computations is removed. There might even be some gain in student enthusiasm if students are, as sometimes rumored, enamored with computers[2]. Finally all this is possible without the student being required to learn any computer programming whatsoever: questions concerning numerical inputs are asked by the computer and the student simply supplies the numbers.

There are several disadvantages although some can be avoided if proper caution is exercised. First, both the student and the instructor may become too problem or number oriented and, consequently, too little emphasis is given to understanding the theory which underlies the numerical results. Secondly, a full range of routines may be desirable and the instructor, at least in the beginning of CAI, must devote some time to the programming of demonstration routines and the preparation of appropriate numerical examples.

Having set the stage for this first use of the computer let us present examples from both microeconomics and macroeconomics.

Case 1: Microeconomics

The first micro calculating routine that we use at Notre Dame is called MI1 (Micro-Instruction1). Sample output from MI1 is illustrated in Figure 1. The program is written in BASIC[3].

The program begins by printing out the available options, or what it can do. The student selects an option; as in Figure 1 supply elasticity. In certain instances the student must select a particular model within the option by specifying what independent variables are included within the model. The model is presented, input is requested, and the answer is given. The student is then given the choice of changing an input within the given model, switching to one of the other available options or terminating the session.

This program with its various options allows several demonstrations. With the quantity demanded option the inverse relationship between price and quantity demand is easily indicated and demand schedules or Engel curves can be readily constructed. Effects of income changes as well as the effects of changes in the prices of complement and substitute commodities on quantity demanded can also be easily indicated. The quantity supplied option seeks to indicate the direct price-quantity relationship for supply. The various elasticity options simply attempt to facilitate computations. The student can also use the various options in conjunction with one another: use the quantity demanded option to generate a demand schedule (or Engel Curves) and then the demand elasticity (or income elasticity) option to calculate the demand elasticity coefficient over various range of the schedule and thereby verify that the demand elasticity coefficient is not necessarily a constant over the entire demand schedule.

Case 2: Macroeconomics

MA1 (Macro Assistance 1) represents our first macroeconomic calculating routine. Sample output is illustrated in Figure 2 and the program is also written in BASIC.

The program begins by printing out five alternative macro models, identified as MA1 through MA5. The student selects a particular model and provides the input requested. The values of the endogenous variables are then calculated and printed. The student then has the choice of modifying a variable within that model, switching to a different model, or terminating the session.

The student by using this program can fully explore the implications of changing the exogenous variables within a model. Then he can switch to another model and fully explore that model. He then can compare differences in results for the two models; that is, isolate the effects in terms of modified numerical results of similar exogenous variable changes between the two models.

The Computer as a Teaching Assistant

There has been a continuous advancement in the role played by the computer and optical scanning equipment in the grading of objective examinations[4]. But the computer can do even more within this particular education context. It can be programmed to ask questions or set up

MII

```

RUN
THIS PROGRAM CALCULATES THE FOLLOWING: QUANTITY DEMANDED,
QUANTITY SUPPLIED, DEMAND ELASTICITY, SUPPLY ELASTICITY,
INCOME ELASTICITY, AND CROSS ELASTICITY.
INPUT NON-NUMERIC DATA WITH QUOTATION MARKS.
WHAT WOULD YOU LIKE TO CALCULATE
? "SUPPLY ELASTICITY"
SUPPLY ELASTICITY EQUATION IS:
E=((Q2-Q1)/((Q2+Q1)/2))/((P2-P1)/(P2+P1)/2))
WHAT IS THE ORIGINAL QUANTITY SUPPLIED
? 12550
WHAT IS THE NEW QUANTITY SUPPLIED
? 10459
WHAT IS THE ORIGINAL PRICE
? 200
WHAT IS THE NEW PRICE
? 189
SUPPLY ELASTICITY IS:
DO YOU WISH TO CHANGE SOMETHING
? "YES"
WHAT IS THE ORIGINAL QUANTITY SUPPLIED
? 12550
WHAT IS THE NEW QUANTITY SUPPLIED
? 12670
WHAT IS THE ORIGINAL PRICE
? 200
WHAT IS THE NEW PRICE
? 199
SUPPLY ELASTICITY IS:
DO YOU WISH TO CHANGE SOMETHING
? "NO"
DO YOU WISH TO CALCULATE SOMETHING ELSE
? "YES"
WHAT WOULD YOU LIKE TO CALCULATE
? "CROSS ELASTICITY"
CROSS ELASTICITY EQUATION IS:
E=((Q2-Q1)/((Q2+Q1)/2))/((P2-P1)/((P2+P1)/2))
WHAT IS THE ORIGINAL QUANTITY
? 1200
WHAT IS THE NEW QUANTITY
? 1350
WHAT IS THE ORIGINAL PRICE OF THE OTHER GOOD
? 35
WHAT IS THE NEW PRICE OF THE OTHER GOOD
? 45
THE TWO GOODS ARE SUBSTITUTES
CROSS ELASTICITY IS: 0.47
DO YOU WISH TO CHANGE SOMETHING
? "NO"
DO YOU WISH TO CALCULATE SOMETHING ELSE
? "NO"
EDIT

```

FIGURE 1.

MA1

```

EDIT CL(MA1) BASIC
EDIT RUN
THIS PROGRAM MAKES CALCULATIONS FOR
FIVE DIFFERENT MACROECONOMIC MODELS.
MODEL M1 APPEARS AS:
Y = C + I
C = A + (B*Y)
I = I
MODEL M2 APPEARS AS:
Y = C + I + G
C = A + (B*Y)
I = I
G = G
MODEL M3 APPEARS AS:
Y = C + I
C = A + (B*Y)
I = D - (E*R)
MODEL M4 APPEARS AS:
Y = C + I + G
C = A + B *(Y - T1 + T2)
I = D - (E*R)
G = G
T1 = T1
T2 = T2
MODEL M5 APPEARS AS:
Y = C + I + G
C = A + B *(Y - T1 + T2)
I = D - (E * R)
G = G
T1 = H + (J * Y)
T2 = T2
WHICH MACRO MODEL DO YOU WISH TO USE?
? "M1"
WHAT IS AUTONOMOUS CONSUMPTION
? 100
WHAT IS THE MARGINAL PROPENSITY TO CONSUME
? .78
WHAT IS AUTONOMOUS INVESTMENT
? 75
THE MULTIPLIER IS 4.545453
EQUILIBRIUM INCOME IS 795.4541
EQUILIBRIUM CONSUMPTION IS 720.4541
DO YOU WISH TO CHANGE SOMETHING
? "NO"
DO YOU WISH TO USE A DIFFERENT MODEL
? "NO"
EDIT END
READY

```

FIGURE 2.

15

problems which the student must solve in a manner that becomes progressively more advanced. In such routines the computer can immediately tell the student whether he is right or wrong and if he is wrong the computer can supply additional information or ask additional questions which will indicate to the student why his initial response was incorrect. In this way the computer acts as a real teaching assistant and not merely as a test corrector.

If a series of objective questions are used, the student simply calls out the routine while sitting at the computer terminal. The computer prints out the initial question which the student answers. If the answer is correct the computer proceeds to the next question. If the answer is incorrect the computer moves into a subroutine which asks the student further questions or provides further information so that the student can understand why his initial response was incorrect and why the correct answer is what it is.

To achieve maximum effectiveness using this approach, the student probably should cover certain assigned materials before using these programs. In this way the student is tested on what he knows in a review situation. Moreover, there should be relatively close coordination between test routine material, course content, and actual test material. Only if this is the case will the student perceive some positive benefit from the use of the test routines.

The advantages of the objective question routine are very much the same as the advantages of any test except the student is in no way penalized for poor performance and there is immediate instruction on any incorrectly answered question. The disadvantages are that the student might become too machine-oriented and indeed might rely too heavily on the test routines as a means of achieving satisfactory grades. A further disadvantage is that a great deal of time is necessary to construct this type of routine especially when the objective question takes the form of a multiple choice question. This might involve an instruction subroutine for each incorrect foil.

If problems are used rather than objective questions some new considerations emerge but the operational elements remain much the same. The student calls out the appropriate routine which contains the problems. If a student gives a correct answer to the initial problem another and more difficult problem is presented. If an incorrect response is given the computer asks the student additional questions or provides further information so that the student can understand why his initial response was incorrect and why the correct answer is what it is.

The problem approach has several advantages. First it is quite amenable to a policy format which students seem to find interesting. Again using the context of an intermediate macroeconomic theory course, models are employed which allow for both monetary and fiscal policy. Thus the student might be given a problem based on such a model where he is asked to manipulate policy in order to achieve a given goal; say, make aggregate demand equal to aggregate supply. The student then attempts to make the necessary policy adjustment. A second advantage of the problem approach is that it may reduce the time required for the preparation of test or review routines.

Perhaps it is advisable to make several comments regarding the advantages of the computer in this pretest or review context vis-a-vis problem sets or workbooks. The computer approach is much more flexible. First numerical problems can be processed more quickly electronically. Secondly the computer routines allow for instant instruction—the follow-up questions or information which lead the student to the correct answer if he makes a mistake. Thirdly because a different environment is used and if care is taken in the construction of the routine, the review can be less tedious. And again there is no requirement that the student acquire any programming knowledge whatever.

Case 1: Microeconomics

MI4 is a general objective question routine. Sample output from MI4 is illustrated in Figure 3. The program is written in BASIC.

The program is a review of supply and demand and focuses on the factors which affect demand, the factors which affect supply, and the effects of changes in demand and supply. The routine includes several multiple choice questions and some true or false and yes or no questions. In each instance where a student gives an incorrect answer, he is given information which will help him get the correct answer and the question is repeated. The student cannot select particular questions but must proceed through the entire ten question routine. The information provided at the very beginning of the program is a review and is considered sufficient for the correct answering of all the questions.

The dialog from programs MI6 and MI7 shown in Figures 4 and 5 represent a somewhat different use of the "tutorial" program approach.

M14

EDIT M14 BASIC
EDIT RUN

THIS PROGRAM IS A MICRO REVIEW ROUTINE.
YOU ARE GIVEN A MARKET MODEL AND TESTED
TO DETERMINE WHETHER OR NOT YOU CAN
MANIPULATE CERTAIN VARIABLES IN THE MODEL IN
ORDER TO ACHIEVE GIVEN OBJECTIVES.

THE DEMAND EQUATION IS:
$$Q1=(A-(B*P)+(C*Y)-(D*P1)+E*P2))/P$$
WHERE
A=CONSTANT TERM
B=COEFFICIENT WHICH REPRESENTS THE WAY IN WHICH
Q1 CHANGES WHEN OWN PRICE CHANGES
P=OWN PRICE
C=COEFFICIENT WHICH EXPRESSES THE WAY IN WHICH
Q1 CHANGES WHEN Y CHANGES
Y=INCOME
D=COEFFICIENT WHICH EXPRESSES THE WAY IN WHICH
Q1 CHANGES WHEN THE PRICE OF COMPLIMENTS CHANGE
P1=PRICE OF COMPLIMENT
E=COEFFICIENT WHICH EXPRESSES THE WAY Q1
CHANGES WHEN THE PRICE OF SUBSTITUTES CHANGE
P2=PRICE OF SUBSTITUTES

THE SUPPLY EQUATION IS:
$$Q2=(F + (G*P)-(H*P1))/P$$
WHERE
Q2=QUANTITY SUPPLIED,
F=CONSTANT TERM
G=COEFFICIENT WHICH EXPRESSES THE WAY Q2 CHANGES
WHEN OWN PRICE CHANGES,
P=OWN PRICE,
H=COEFFICIENT WHICH EXPRESSES THE WAY Q2 CHANGES
WHEN PRICE OF INPUTS CHANGE,
P1=THE PRICE OF INPUTS.

THE FIRST QUESTION IS:
IF YOU WISH TO INCREASE MARKET PRICE AND
QUANTITY, WHAT WOULD YOU DO?
ANSWER 'INCREASE DEMAND', 'DECREASE DEMAND',
'INCREASE SUPPLY', OR 'DECREASE SUPPLY'
?

"INCREASE DEMAND"
YOUR ANSWER IS CORRECT

THE SECOND QUESTION IS:
IF INCOME INCREASES WHAT HAPPENS TO DEMAND?
ANSWER 'INCREASES' OR 'DECREASES'.
?

"DECREASES"
YOUR ANSWER IS INCORRECT. THE RELATIONSHIP
BETWEEN INCOME AND DEMAND IS DIRECT.

THE SECOND QUESTION IS:
IF INCOME INCREASES WHAT HAPPENS TO DEMAND?
ANSWER 'INCREASES' OR 'DECREASES'.
?

FIGURE 3.

EDIT ~~RUN~~ EDIT MI6 BASIC

THIS PROGRAM EXAMINES THE BASIC IDEAS OF A PRODUCTION FUNCTION,
DIMINISHING RETURNS AND THE STAGES OF PRODUCTION.
DO YOU WISH TO HAVE THESE TERMS DEFINED?
ANSWER "YES" OR "NO" .

? "YES"

A PRODUCTION FUNCTION IS A TECHNICAL RELATIONSHIP THAT
INDICATES THE AMOUNT OF OUTPUT THAT CAN PHYSICALLY
BE PRODUCED BY EACH FACTOR OF PRODUCTION OR BY EACH
SET OF FACTORS OF PRODUCTION--I.E., LAND, LABOR, CAPITAL
OR MANAGERIAL ABILITY.

A PRODUCTION FUNCTION IS ALWAYS DEFINED FOR A GIVEN
STATE OF TECHNOLOGY. THE USUAL FACTOR THAT ESTABLISHES
THE STATE OF TECHNOLOGY IS THE CAPITAL OR PLANT AND
EQUIPMENT. ONCE THE INVESTMENT IS MADE IN EQUIPMENT IT
IS FINANCIALLY PROHIBITIVE TO INVEST IN NEWER EQUIPMENT
UNTIL THAT GIVEN PLANT AND EQUIPMENT IS PAID FOR.

DIMINISHING RETURNS MEANS THAT AN INCREASE IN
THE VARIABLE FACTORS OF PRODUCTION (AS INPUTS) WILL INCREASE
OUTPUT. HOWEVER, AFTER SOME POINT IN PRODUCTION THE
INCREASE IN OUTPUT FROM EACH SUCCESSIVE EQUAL INCREASE IN
THE VARIABLE INPUT WILL BE LESS AND LESS BECAUSE OF THE
FIXED FACTORS.

ECONOMIES OF SCALE ASSUMES A GIVEN STATE OF TECHNOLOGY
AND REFERS TO THE RELATIONSHIP OF INPUT CHANGES TO
CHANGES IN OUTPUT.

ASSUME A GIVEN PERCENTAGE CHANGE IN ALL INPUTS.
IF OUTPUT CHANGES BY THE SAME PERCENT WE REFER TO
CONSTANT RETURNS TO SCALE. IF OUTPUT CHANGES BY A
SMALLER AMOUNT OR PERCENT WE REFER TO DECREASING RETURNS TO SCALE
IF OUTPUT CHANGES BY A LARGER AMOUNT, INCREASING
RETURNS TO SCALE WILL OCCUR.

DO YOU WISH THE PRODUCTION FUNCTION TO BE DEFINED?
ANSWER "YES" OR "NO" .

? "YES"

A GENERAL PRODUCTION FUNCTION MAY BE EXPRESSED:
OUTPUT IS A FUNCTION OF LAND INPUTS, LABOR INPUTS, CAPITAL INPUTS, ENTREP
RENEURIAL INPUTS AND A TECHNOLOGY FACTOR. EACH
ITEM CAN BE REPRESENTED BY A SYMBOL:

RESPECTIVELY LAND = L, LABOR = N, CAPITAL = C, AND
ENTREPRENEUR = E AND TECHNOLOGY = T.

THUS OUTPUT = T(L,N,C,E) .

TO MOVE FROM A GENERAL TO A SPECIFIC PRODUCTION FUNCTION
DETAIL MUST BE ADDED TO INDICATE THE EXACT NATURE
OF THE RELATIONSHIP BETWEEN OUTPUT AND THE INPUTS.

A PRODUCTION FUNCTION MIGHT BE AS FOLLOWS:

OUTPUT = T(N,C) .

OUTPUT = T(N,K) WHERE K IS THE FIXED INPUT SET.
WHAT IS THE INITIAL INPUT LEVEL OF N (LABOR)

? 200

A NUMBER BETWEEN 1 AND 5 WILL INDICATE A LOW LEVEL
OF TECHNOLOGY, BETWEEN 60 - 100, A HIGH LEVEL.

WHAT IS THE VALUE OF THE TECHNOLOGICAL FACTOR?

? 5

FIGURE 4.

KI6

OUTPUT = T(N,K) WHERE K IS THE FIXED INPUT SET.
WHAT IS THE INITIAL INPUT LEVEL OF N (LABOR)

? 200

A NUMBER BETWEEN 1 AND 5 WILL INDICATE A LOW LEVEL
OF TECHNOLOGY, BETWEEN 60 - 100, A HIGH LEVEL.

WHAT IS THE VALUE OF THE TECHNOLOGICAL FACTOR?

? 60

A NUMBER BETWEEN .1 AND .5 WILL INDICATE A
AUTOMATED OR HIGH FIXED FACTOR FIRM. A NUMBER BETWEEN
1.0 AND 2.0 WILL INDICATE A SEMI-AUTOMATED FIRM OR
A LOW FIXED FACTOR FIRM.

? .3

LEVEL OF PRODUCTION	TOTAL VARIABLE INPUT	TOTAL OUTPUT	AVERAGE PRODUCT	MARGINAL PRODUCT
1	200	80000.	400.00	400.00
2	220	176000.	800.00	4800.00
3	240	432000.	1800.00	12800.01
4	260	831999.	3200.00	20000.18
5	280	1399997.	5000.00	28400.23
6	300	2939992.	9800.00	77000.69
7	320	3135090.	9800.00	9800.02
8	340	3331988.	9800.00	9800.02
9	360	3527985.	9800.00	9799.97
10	380	3532884.	9297.10	244.95
11	400	3533006.	8832.55	6.10
12	420	3533009.	8411.97	0.15
13	440	3533009.	8029.61	0.00

WHAT IS THE TOTAL FIXED COST OF THE PLANT, OR FIXED COST OF THE FACTORS.

? 150000

WHAT IS THE COST OF HIRING EACH VARIABLE INPUT FOR THE PERIOD?

? 500

FIGURE 4. Continued.

MI6

LEVEL OF PRODUCTION	TOTAL OUTPUT	TOTAL VARIABLE COST	TOTAL FIXED COST	TOTAL COST
1	80000.	100000.	150000.	250000.
2	176000.	110000.	150000.	260000.
3	432000.	120000.	150000.	270000.
4	831999.	130000.	150000.	280000.
5	1399997.	140000.	150000.	290000.
6	2939992.	150000.	150000.	300000.
7	3135990.	160000.	150000.	310000.
8	3331988.	169999.	150000.	319999.
9	3527985.	179999.	150000.	329999.
10	3532884.	189999.	150000.	339999.
11	3533006.	199999.	150000.	349999.
12	3533009.	209999.	150000.	359999.
13	3533009.	219999.	150000.	369999.

LEVEL OF PRODUCTION	AVERAGE TOTAL COST	AVERAGE VARIABLE COST	AVERAGE FIXED COST	MARGINAL COST
1	3.125	1.250	1.875	1.250
2	1.477	0.625	0.852	0.104
3	0.625	0.278	0.347	0.039
4	0.337	0.156	0.180	0.025
5	0.207	0.100	0.107	0.018
6	0.102	0.051	0.051	0.006
7	0.099	0.051	0.048	0.051
8	0.096	0.051	0.045	0.051
9	0.094	0.051	0.043	0.051
10	0.096	0.054	0.042	2.041
11	0.099	0.057	0.042	81.966
12	0.102	0.059	0.042	3333.292

DO YOU WISH TO CHANGE SOMETHING?
 ANSWER "INPUT" OR "COST" .
 7 "NO"
 EDIT

FIGURE 4. Continued.

Program MI6 constructs and defines a particular production function. It then develops the physical product schedules (Total Product, APP, and MPP). Next, fixed and variable cost input allows the development of the total and per unit cost schedules.

Program MI7 is a set of 15 questions, shown in Figure 4, keyed on the output from MI6. If the student gives an incorrect answer, he is provided additional information and the question is asked again. This demonstration is keyed to the intermediate microeconomic book by R. Leftwich.

Case 2: Macroeconomics

Our most elementary macro policy routine is called GN15. Figure 6 is sample output. The program is also written in BASIC.

In this case it is presumed that the instructor has provided an explanation of the underlying macro model. If the student makes an incorrect policy decision he is simply told that he is incorrect. Further information or explanation is not provided because the only ingredients necessary for a correct decision, the difference between 'GNP' and 'Full Employment GNP' and the value of the "multiplier," are printed out once the student supplies the initially requested input.

The dialog from program MA3 is shown in Figure 7. It is designed to illustrate the IS and LM analysis. It is useful when used with the graphic IS and LM analysis.

The Computer as a "Real World" Environment for the Testing of Alternative Hypotheses

In many disciplines the major objectives of instruction are to equip students with tools of analysis and to develop within the student an ability to apply these tools in examining and solving problems. When the computer functions as a "real world" environment for the testing of alternative hypotheses, it is providing the student an opportunity to apply tools in examining and solving problems. Such routines have certain basic characteristics, primarily in terms of format. First the routine provides the student with basic data—the initial state of the world. Secondly the student acting as a decision maker must manipulate certain variables in order to achieve specific goals. This manipulation of variables is based on student's theoretical knowledge as reinforced or supplemented by his examination of the basic data. Thirdly a new set of data emerges which embodies the results of the student's initial manipulation of variables. At this stage the student can ascertain the results of his decision and his success as a decision maker. The routine can provide for further decisions with further data output or it can terminate the program after only a single set of decisions.

There are of course many differences between such routines. One difference is in terms of the number of decision makers: the individual versus the group game. In the group game each of several students or several groups of students manipulates or makes decisions for a single firm or a single country. The second set of data which emerges contains the results of all these different decisions. Thus the student is competing against other students and must in some cases anticipate their decisions. In such instances batch processing is required. If only a single individual is involved there is no simultaneous interaction of decisions so that the second set of data simply reflects the decisions of the single student. In this case time sharing can be used.

Another difference is in terms of the types of changes which occur between stages. In the simpler versions the only source of change is the decision of the student as caused by a change in a policy variable. In more complex versions there is systematic variation in non-policy exogenous variables as well. Indeed the student is usually required to recognize such changes if he is to make correct policy decisions. Finally there are versions in which a third type of change, random change, is added. This is usually included for the sake of realism; that is, the future is not perfectly predictable and consequently even the most carefully designed policy can be frustrated by random change.

These routines also vary in other respects including the number of goals to be achieved, the number of policy variables over which the student has control, and the compatibility of goals. Some routines even include statistical sub-routines that the student can use in examining and estimating behavioral relationships within the "real world" of the routine.

To fully exploit the intellectual potential of this role of the computer is extremely difficult. First the routine must be constructed so that the student cannot achieve success by luck but at the same time allow for experimentation which will lead to understanding. Secondly, the real world contained within the routine must bear some similarity that basic behavioral patterns are immediately obvious. Thirdly, if statistical procedures must be used to estimate the magnitude of certain relationships, the student should have some basic understanding of these procedures.

MIJN

THIS TUTORIAL PROGRAM IS DESIGNED TO BE USED WITH PROGRAM MI6 ON PRODUCTION AND COST FUNCTIONS. DO YOU WISH THE DEFINITIONS AND INSTRUCTIONS FOR THIS PROGRAM? ANSWER: "YES" OR "NO".

? "YES"

THE TERM 'LEVEL OF PRODUCTION' IS USED IN THIS PROGRAM TO REFER TO THE TABLE OUTPUT COLUMNS OBTAINED FROM MI6.

YOU WILL NEED THE OUTPUT FROM MI6 IN WORKING THIS PROGRAM.

YOU WILL ALSO NEED THE BOOK, 'PRICE SYSTEM AND RESOURCE ALLOCATION' BY R. LEFTWICH.

WHEN AN ANSWER IS REQUIRED THAT RELATES TO THE LEVEL OF PRODUCTION YOU MUST BE CAREFUL TO PROVIDE ONLY A SINGLE PRODUCTION LEVEL, OR A BEGINNING AND ENDING PRODUCTION LEVEL AS REQUIRED BY EACH QUESTION.

QUESTION 1:

DOES "DIMINISHING RETURNS TO SCALE" MEAN THE SAME THING AS "DIMINISHING TOTAL PRODUCT INCREASES"?

ANSWER: "YES" OR "NO".

? "YES"

YOUR ANSWER IS CORRECT.

QUESTION 2:

IS THE LAW OF DIMINISHING RETURNS MEASURED ON THE AVERAGE OR MARGINAL PRODUCT SCHEDULE?

ANSWER 'AVERAGE' OR 'MARGINAL'.

? "AVERAGE"

YOUR ANSWER IS INCORRECT. SEE LEFTWICH PP.117,119.

QUESTION 2:

IS THE LAW OF DIMINISHING RETURNS MEASURED ON THE AVERAGE OR MARGINAL PRODUCT SCHEDULE?

ANSWER 'AVERAGE' OR 'MARGINAL'.

? "MARGINAL"

YOUR ANSWER IS CORRECT.

QUESTION 3:

WHAT LEVELS OF PRODUCTION INDICATE INCREASING RETURNS TO SCALE?

? 2,7

YOUR ANSWER IS INCORRECT. RECHECK THE SCHEDULE.

SEE LEFTWICH, P.119, FIGURE 7-1.

QUESTION 3:

WHAT LEVELS OF PRODUCTION INDICATE INCREASING RETURNS TO SCALE?

? 1,6

FIGURE 5.

157

GNI

EDIT CL(GNI) BASIC
EDIT RUN

THIS IS A SIMPLE MULTIPLIER MODEL STUDY OF EMPLOYMENT THEORY.
WHEN THE PROGRAM ASKS FOR INPUT THAT
IS IN THE FORM OF A NUMBER, INCLUDE THE DECIMAL POINT AND
THE APPROPRIATE SIGN AS REQUIRED FOR EACH INPUT SITUATION.
WHEN ALPHABETIC INPUT IS REQUIRED BE SURE TO ENCLOSE
THE INPUT IN QUOTATION MARKS

WHAT IS AUTONOMOUS CONSUMPTION
? 100
WHAT IS THE PROPENSITY TO CONSUME OUT OF GNP
? .78
WHAT IS GOVERNMENT DEMAND FOR GOODS AND SERVICES
? 150
WHAT IS INVESTMENT DEMAND
? 175

THE MULTIPLIER IS 4.545453
GNP IS 1931.817
FULL EMPLOYMENT GNP IS 1738
TO REACH FULL EMPLOYMENT, BY HOW MUCH SHOULD GOVERNMENT DEMAND CHANGE
? -43
YOU ARE CORRECT
WOULD YOU LIKE TO CHANGE SOMETHING?
ANSWER "YES" OR "NO".
? "YES"
AUT CON, MPC, I, OR G
? "MPC"
WHAT IS THE NEW VALUE FOR MPC
? .85

THE MULTIPLIER IS 6.666666
GNP IS 2546.666
FULL EMPLOYMENT GNP IS 2724
TO REACH FULL EMPLOYMENT, BY HOW MUCH SHOULD GOVERNMENT DEMAND CHANGE
? 30
YOUR ANSWER IS WRONG; THAT WOULD CAUSE GNP TO BE 2746.666
TO REACH FULL EMPLOYMENT, BY HOW MUCH SHOULD GOVERNMENT DEMAND CHANGE
? 29
YOUR ANSWER IS WRONG; THAT WOULD CAUSE GNP TO BE 2739.999
TO REACH FULL EMPLOYMENT, BY HOW MUCH SHOULD GOVERNMENT DEMAND CHANGE
? 28
YOUR ANSWER IS WRONG; THAT WOULD CAUSE GNP TO BE 2733.333
TO REACH FULL EMPLOYMENT, BY HOW MUCH SHOULD GOVERNMENT DEMAND CHANGE
? 27
YOU ARE CORRECT
WOULD YOU LIKE TO CHANGE SOMETHING?
ANSWER "YES" OR "NO".
? "NO"
EDIT

FIGURE 6.

158

147

MA3

EDIT CL(MA3) BASIC
RUN

THIS PROGRAM COMPUTES VALUES FOR A MACRO ECONOMY THAT CONSISTS ONLY OF A COMMODITY MARKET AND A MONEY MARKET; THAT IS, THE ECONOMY CAN BE REPRESENTED BY IS AND LM CURVES.

THIS PROGRAM IS THE SIMPLER OF IS AND LM MODELS. THE OTHER PROGRAM IS LM2 THE COMMODITY MARKET OR IS PORTION OF THIS MODEL CONSISTS OF THE FOLLOWING EQUATIONS:
COMMODITY MARKET EQUILIBRIUM
CONSUMPTION FUNCTION
DISPOSABLE INCOME

Y=C+I+G

YI=Y-T

T=T1

I=I1-I2R

G=G1

THE MONEY MARKET OR LM PORTION OF THIS MODEL CONSISTS OF THE FOLLOWING EQUATIONS:
MONEY MARKET EQUILIBRIUM
DEMAND FOR MONEY
SUPPLY OF MONEY

MD=M1+M2Y-M3R

MS=M9

DO YOU WISH TO HAVE EACH VARIABLE DEFINED?

ANSWER 'YES' OR 'NO'

? "YES"

? "NO"

Y=INCOME, C=CONSUMPTION, I=INVESTMENT, G= GOVERNMENT EXPENDITURES, C1=AUTONOMOUS CONSUMPTION EXPENDITURES, C2=MARGINAL PROPENSITY TO CONSUME OUT OF DISPOSABLE INCOME, YI=DISPOSABLE INCOME, T=TAXES, I1=AUTONOMOUS INVESTMENT, I2=INVESTMENT-INTEREST RATE COEFFICIENT, MS=SUPPLY OF MONEY, MD=DEMAND FOR MONEY, M1=CONSTANT TERM IN DEMAND FOR MONEY, M2=INCOME -DEMAND FOR MONEY COEFFICIENT, AND M3= INTEREST RATE-DEMAND FOR MONEY COEFFICIENT.

THE THREE EXOGENOUS POLICY VARIABLES ARE: T OR TAXES, G OR GOVERNMENT EXPENDITURES, AND MS OR THE MONEY SUPPLY,

WHAT IS THE VALUE OF T?

? 40.

WHAT IS THE VALUE OF G?

? 200

WHAT IS THE VALUE OF MS?

? 215

MA3

THE REMAINING EXOGENOUS VARIABLES ARE: C1 OR AUTONOMOUS CONSUMPTION, C2 OR THE MARGINAL PROPENSITY TO CONSUME, I1 OR AUTONOMOUS INVESTMENT, I2 OR THE INVESTMENT INTEREST RATE COEFFICIENT, M1 OR THE CONSTANT TERM IN THE DEMAND FOR MONEY, M2 OR THE DEMAND FOR MONEY + INCOME COEFFICIENT, M3 OR THE DEMAND FOR MONEY INTEREST RATE COEFFICIENT.

WHAT IS THE VALUE OF C1?

? 40.

WHAT IS THE VALUE OF C2?

? .7

WHAT IS THE VALUE OF I1?

? 30.

WHAT IS THE VALUE OF I2?

? .005

WHAT IS THE VALUE OF M1?

? 10.

WHAT IS THE VALUE OF M2?

? .25

WHAT IS THE VALUE OF M3?

? .1

THE ENDOGENOUS VARIABLES ARE Y OR INCOME, Y1 OR DISPOSABLE INCOME, R OR THE INTEREST RATE, C OR CONSUMPTION, AND I OR INVESTMENT, THE VALUES OF THESE VARIABLES ARE:

Y IS 877.3909

Y1 IS 837.3909

C IS 626.1736

I IS 29.28261

R IS (INTEREST RATE) 1.434770

THE POLICY MULTIPLIERS ARE K OR THE

GOVERNMENT EXPENDITURES MULTIPLIER, K1 THE

GOVERNMENT TAX MULTIPLIER, AND K2 THE

MONEY MULTIPLIER. THE VALUE OF THESE

MULTIPLIERS ARE:

K IS 3.478260

K1 IS -2.434782

K2 IS .1739130

DO YOU WISH TO CHANGE SOMETHING?

ANSWER 'YES' OR 'NO'

? "NO"

EDIT

The advantages of such routines lie primarily in the experience which the student obtains in attempting to solve problems on the basis of his knowledge and ability. The demand to make material relevant is frequently a demand to use knowledge in solving current real problems. Thus by the use of the computer in this way not only does the student gain experience in an area in which experience is obtained only with difficulty if at all, but at the very same time demonstrates for himself the necessity and usefulness of theoretical knowledge[6].

The major disadvantages reside primarily in setting up the routine so as to achieve maximum intellectual gain. If a number of students are making decisions simultaneously and batch processing is required, the instantaneous communication between student and computer is eliminated. If there is no interaction of student decisions then some realism is lost. At Notre Dame we have both types of programs.

CASE 1: Microeconomics

A case where microeconomics can effectively simulate a real world situation is in a market type relationship where the student must act as if he were a part of an overall business environment. AGSIM[7] is such a program. It is a general simulation that involves two elements: (1) a specified macro economic model and (2) students acting as competitive business firms within a competitive market in an attempt to earn a profit.

The output from AGSIM is presented in Figure 8. The program is written in FORTRAN. In this program the instructor established the macro model, he sets variables like the average propensity to consume, the federal tax rate, government purchases per firm, and government transfers per firm. The price of the good is also set by the instructor. Each student acting as a firm makes his own decisions in each period regarding sales and investment. The results are calculated in the context of both the specified macro model and the competitive market which is represented by the interaction of decisions by all students. The student gets computer output for each successive period. He must examine this output before making decisions for the next period. The teacher may periodically change the macro environment or he may modify the "competitive market price."

CASE 2: Macroeconomics

Price stability and full employment are explicit goals of the American economy. With GN6 we have attempted to create a situation where the student can manipulate both monetary and fiscal policy in order to achieve both of these goals. One problem is that the two goals are not fully consistent with one another; zero price change and zero unemployment are not simultaneously possible. Another aspect of realism is a continuous growth in the productive capability of the economy.

Figure 9 represents output for program GN6. We can follow the program as it operates for the student[8]. The program asks if instructions are desired, and if they are, the information is printed out. Next the program asks if the student desires to see the model, which is printed out if desired. Then the student puts in each of the four variables as shown, ten quarters for each variable. To control the variables, he simply changes the magnitude of each variable in successive quarters. The answers are printed out as shown in the last half of Figure 9. After each set of output the student may branch back to change any variable, or he may end the program. The program is written in BASIC. The instructor usually sets the policy goals for the student, i.e., GNP, full employment GNP, unemployment 6%, price change .6 per quarter, etc.

Conclusion

This represents then the status of our attempt to develop a complete system of CAI within Economics. Our objectives over the immediate future are to fill in gaps in terms of available programs and to obtain more experience with the system within the classroom setting. Once this is accomplished we can begin to test whether or not CAI is an effective educational tool.

Even though we are only at the initial stage of CAI we feel that aside from current resource limitations, the use of the computer as an instructional device is limited only by our own imaginations. Hopefully this paper by pointing out and exploring alternative educational roles for computers will stimulate and encourage further investigation and experimentation with CAI.

AGSIM

PERFECT COMPETITION, APC = .75, TAX RATE = .14

24 NOV 71 PAGE 8

INITIATED BY BUSINESS CONDITIONS SIMULATION (AGSIM)

RESULTS OF SIMULATION FOR THE INTERMEDIATE ECONOMIC THEORY CLASS

PERIOD = 4, INTEREST = .14, TAX RATE = .14, TRANSFER = 2640.00, GOVERNMENT PURCHASES = 3300.00, APC = .75

ENTERPRISE'S CHOICE IN MAKING NEXT PERIOD DECISIONS

FIRM	PRD.	INVEST.	SALES	WAGES	(F-UNG INVENTORY	= INVENTORY) COST	FIXED COST	PROFIT
CHAMBERS PLUMBING	105.0	.0	1462.93	1472.20	15.9	* 3 = 47.50	399.00	44.17
YOUR CAPACITY FOR NEXT PERIOD IS		133.0						
GALL'S CONSTRUCTION	8.0	.0	2997.50	2173.12	.0	* 3 = .00	393.00	331.38
YOUR CAPACITY FOR NEXT PERIOD IS		131.0						
HUTLER WHITE CO.	59.0	.0	1962.93	1472.20	19.9	* 3 = 59.50	396.00	35.17
YOUR CAPACITY FOR NEXT PERIOD IS		132.0						
MOSCHES CO.	59.0	.0	1462.93	1472.20	6.9	* 3 = 20.50	393.00	71.17
YOUR CAPACITY FOR NEXT PERIOD IS		131.0						
J. T. LAFFLE CO.	85.0	.0	1962.93	1472.20	5.9	* 3 = 17.50	399.00	74.17
YOUR CAPACITY FOR NEXT PERIOD IS		133.0						
POEPE WRIGHT CO.	80.0	.0	1962.93	1472.20	22.9	* 3 = 68.50	396.00	26.17
YOUR CAPACITY FOR NEXT PERIOD IS		132.0						
BERGDA MALT LICKER	130.0	.0	1962.93	1472.20	74.9	* 3 = 224.50	396.00	-129.83
YOUR CAPACITY FOR NEXT PERIOD IS		132.0						
M. J. GAYLE CORP.	74.0	.0	2379.00	1784.25	.0	* 3 = .00	396.00	198.75
YOUR CAPACITY FOR NEXT PERIOD IS		132.0						
SUGAR INDUSTRIES	10.0	40.0	1962.93	1472.20	42.9	* 3 = 128.50	402.00	-39.83
YOUR CAPACITY FOR NEXT PERIOD IS		135.0						
LACIE WHITE CO.	50.0	.0	1962.93	1472.20	6.9	* 3 = 20.50	414.00	56.17
YOUR CAPACITY FOR NEXT PERIOD IS		130.0						
HETTS LIGHTS	40.0	.0	2582.00	1721.50	.0	* 3 = .00	393.00	241.50
YOUR CAPACITY FOR NEXT PERIOD IS		133.0						
KAISER INTERNATIONAL	55.0	.0	1962.93	1472.20	14.9	* 3 = 44.50	393.00	53.17
YOUR CAPACITY FOR NEXT PERIOD IS		131.0						
ROGERS CORP.	80.0	.0	2606.00	2104.50	.0	* 3 = .00	393.00	308.50
YOUR CAPACITY FOR NEXT PERIOD IS		131.0						
WILSON	59.0	.0	1962.93	1472.20	12.9	* 3 = 38.50	393.00	68.17
YOUR CAPACITY FOR NEXT PERIOD IS		132.0						

FIGURE 8.

161

AGRID

PERFECT COMPETITION, APC = .75, TAX RATE = .14

29 NOV 71 PAGE 8

YOUR CAPACITY FOR NEXT PERIOD IS	131.0							
MCERMILLAN CORP	100.0	40	1962.93	1472.20	7.9	0.3 = 23.56	390.00	7.17
YOUR CAPACITY FOR NEXT PERIOD IS	130.0							
MCQUANON STAGE SUITS	100.0	40.0	1962.93	1472.20	1.9	0.3 = 5.96	402.00	83.17
YOUR CAPACITY FOR NEXT PERIOD IS	135.0							
MCWELL CORPORATION	80.0	40	1962.93	1472.20	15.9	0.3 = 47.56	402.00	41.17
YOUR CAPACITY FOR NEXT PERIOD IS	134.0							
MILES SYMBOL CO.	100.0	40	1962.93	1472.20	5.9	0.3 = 17.56	393.00	80.17
YOUR CAPACITY FOR NEXT PERIOD IS	131.0							
MONTAGNE SELECT	95.0	40	1962.93	1472.20	11.9	0.3 = 35.56	396.00	59.17
YOUR CAPACITY FOR NEXT PERIOD IS	132.0							
MELTSCHERF GUMMICH	110.0	40	1962.93	1472.20	11.9	0.3 = 35.56	330.00	125.17
YOUR CAPACITY FOR NEXT PERIOD IS	110.0							
SCHLAFERS SHIRVES	70.0	40.0	1962.93	1472.20	32.9	0.3 = 98.56	405.00	-12.83
YOUR CAPACITY FOR NEXT PERIOD IS	136.0							
MCACH UNLIMITED	85.0	40	2206.00	2104.50	.0	0.3 = .00	399.00	302.50
YOUR CAPACITY FOR NEXT PERIOD IS	133.0							
MYERS-COOP	97.0	40	1962.93	1472.20	2.9	0.3 = 8.56	390.00	92.17
YOUR CAPACITY FOR NEXT PERIOD IS	130.0							
MCHE WIGGOT CO.	60.0	80.0	2562.00	1921.50	.0	0.3 = .00	399.00	241.50
YOUR CAPACITY FOR NEXT PERIOD IS	135.0							
PERTESS, INC.	78.0	40	1962.93	1472.20	1.9	0.3 = 5.56	450.00	35.17
YOUR CAPACITY FOR NEXT PERIOD IS	150.0							
MESNEY GARRET CO.	75.0	40.0	2297.50	1715.67	.0	0.3 = .00	399.00	172.88
YOUR CAPACITY FOR NEXT PERIOD IS	132.0							
GRAVEN INCORPORATED	95.0	40	1962.93	1472.20	3.9	0.3 = 11.56	396.00	83.17
YOUR CAPACITY FOR NEXT PERIOD IS	132.0							
FLOOD ENTERPRISES	100.0	40	1962.93	1472.20	61.9	0.3 = 185.56	402.00	-96.83
YOUR CAPACITY FOR NEXT PERIOD IS	134.0							
COO LEFT CORPORATION	140.0	80.0	1962.93	1472.20	65.9	0.3 = 197.56	420.00	-126.83
YOUR CAPACITY FOR NEXT PERIOD IS	142.0							
SCALIAS WIGGOT CO.	40.0	40	1962.93	1472.20	56.9	0.3 = 170.56	456.00	-135.83
YOUR CAPACITY FOR NEXT PERIOD IS	152.0							
JURIST CONSO	35.0	40	1962.93	1472.20	7.9	0.3 = 23.56	396.00	71.17
YOUR CAPACITY FOR NEXT PERIOD IS	132.0							
CEBY DIRECTILES	110.0	40.0	1962.93	1472.20	35.9	0.3 = 119.56	393.00	-21.83
YOUR CAPACITY FOR NEXT PERIOD IS	135.0							
CEBY M.S. ELECTRIC	90.0	40.0	1962.93	1472.20	34.9	0.3 = 104.56	396.00	-9.83
YOUR CAPACITY FOR NEXT PERIOD IS	134.0							

APPROXIMATE DATA FOR ESTIMATING OVERALL BUSINESS CONDITIONS. USE DATA IN ESTIMATING NEXT PERIOD DECISIONS

TOTAL PROFITS ARE 2359.52
 AVERAGE PROFITS ARE 72.71
 TOTAL INVENTORIES ARE 586.18
 AVERAGE INVENTORIES ARE 17.76
 TOTAL CAPACITY IS 4403.00
 AVERAGE CAPACITY IS 133.42
 END OF GRID

FIGURE 8. Continued.

GN6

```
EDIT CL(GN6) BASIC
EDIT RUN
DO YOU WISH INSTRUCTIONS ON HOW TO RUN THE PROGRAM?
ANSWER 'YES' OR 'NO'
? "YES"
DATA MUST BE ENTERED BY VARIABLE. THE FIRST DATA IS
THE AMOUNT OF PERSONAL TAXES. THE SECOND IS THE
AMOUNT OF CORPORATE TAXES. FOLLOWED BY THE MONEY
SUPPLY AND THE AMOUNT OF GOVERNMENT EXPENDITURES.
DATA ARE ENTERED IN A ROW FOR THE TEN QUARTERS.
CORPORATE TAXES OF 45 BILLION WOULD BE ENTERED AS 45
DO YOU WISH TO SEE THE MODEL?
ANSWER 'YES' OR 'NO'
? "YES"
DEFINITIONS AND RELATIONS
GNP = CON + INV + GOV
CON = C + MPC*DI
DI = GNP - CPROF - PTAX
CPROF = H*GNP
INV = I1 + A*RE + G*RATE
RE = CPROF - PTAX
RATE = U*MONEY + V*GNP + 16.0
POTENTIAL OUTPUT COMPUTED FROM A COBB-DOUGLAS
PRODUCTION FUNCTION.
GNPF = TECH*A * LAB^B * CAP^C WHERE A+B+C=1
ENTER DATA FOR PERSONAL TAXES (10 QTRS.)
? 90,90,90,90,90,90,90,90,95,95
ENTER DATA FOR CORPORATE TAXES ( 10 QTRS. )
? 45,45,45,45,45,45,44,43,42,42
ENTER THE MONEY SUPPLY (10 QTRS)
? 200,200,200,200,200,200,200,200,200,200
ENTER DATA FOR GOVERNMENT EXPENDITURES (10 QTRS.)
? 215,215,215,215,215,215,220,220,220,220

QUART
EDIT
```

FIGURE 9.

01/163

GN6

QUARTER	GNP	POTENTIAL GNP	DISPOSABLE INCOME
1	752.9	959.0	587.6
2	752.9	968.9	587.6
3	752.9	978.8	587.6
4	752.9	988.6	587.6
5	752.9	998.4	587.6
6	752.9	1008.2	587.6
7	774.9	1018.1	607.4
8	778.5	1028.1	610.7
9	769.4	1038.1	597.4
10	769.4	1048.0	597.4

CONSUMPTION	INVESTMENT	GOV'T SPENDING	RET. EARNINGS
461.3	76.6	215.0	30.3
461.3	76.6	215.0	30.3
461.3	76.6	215.0	30.3
461.3	76.6	215.0	30.3
461.3	76.6	215.0	30.3
461.3	76.6	215.0	30.3
475.2	79.7	220.0	33.5
477.5	81.0	220.0	34.9
468.2	81.2	220.0	34.9
468.2	81.2	220.0	34.9

INTEREST RATE	PRICE CHANGE	UNEMPLOYMENT
7.904	- 3.447	8.947
7.904	- 3.688	9.188
7.904	- 3.923	9.423
7.904	- 4.152	9.652
7.904	- 4.376	9.876
7.904	- 4.595	10.095
7.930	- 4.167	9.667
7.934	- 4.283	9.783
7.923	- 4.765	10.265
7.923	- 4.976	10.476

DO YOU WISH TO CHANGE A VARIABLE? ANSWER THE VARIABLE NAME IN QUOTES OR 'NO'.

THE VARIABLES ARE "PRIVATE TAXES" "CORPORATE TAXES"
"MONEY SUPPLY" AND "GOVERNMENT EXPENDITURES"

? "NO"

EDIT

FIGURE 9. Continued.

FOOTNOTES

1. For alternative examples of the use of computers in undergraduate programs see Proceedings of a Conference on Computers in the Undergraduate Curricula (Iowa City: Center for Conferences and Institutes, University of Iowa, 1970) and Second Conference on Computers in the Undergraduate Curricula (Hanover: University Press of New England, 1971). For additional examples within Economics see New Developments in the Teaching of Economics edited by K. G. Lunsden (Englewood Cliffs: Prentice-Hall, 1967) and Recent Research in Economic Education edited by K. G. Lunsden (Englewood Cliffs: Prentice-Hall, 1970). See also Davisson, W. Information Processing: Applications in the Social and Behavioral Sciences (New York: Appleton Century Crofts, 1970).
2. See Sharac, J. and D. Russ, "Evaluation of Student Learning and Information Utilization in a Computer Simulated Economic Model" in Second Conference, pp. 94-98.
3. BASIC is a time-sharing computer language.
4. Kel 7, A. C. "The Economics of Teaching: The Role of Tips" in Recent Research, pp. 44-66.
5. Modified from a BASIC Program obtained from George Pilot, Department of Economics, Dartmouth College.
6. Perhaps another advantage is that such routines are on the market. Two examples are: Econometric Gaming: A Kit for Computer Analysis by L. R. Klein and H. K. Evans (New York: The Macmillan Company, 1969) and Macro: A Game of Growth and Policy by Peter Lindert (New York: Holt, Rinehart and Winston, 1970).
7. Modified from a FORTRAN program supplied by Professor Bonney, Luther College, Iowa.
8. Modified from a FORTRAN program supplied by Professor Bonney, Luther College, Iowa.

INTEGRATING COMPUTER PROGRAMS IN ECONOMICS VIA TIME SHARING TERMINALS

Frank DeFelice
Belmont Abbey College
Belmont, North Carolina 28012
Telephone: (704) 825-3711

Introduction

Simulations of the various functional areas of business based on complex interactive models are used almost routinely at large universities, and hardly at all at small liberal arts colleges. The universities have their own large computers and computer personnel whereas the typical small liberal arts college may have a terminal in a time-sharing system and a part-time faculty member as terminal manager. Given these differences in hardware and back-up personnel, it is easy to see why small colleges are not making much use of the complex interactive business simulations. However, it is technically feasible, and not really very difficult, to run these simulations over low speed time-sharing terminals if two conditions are met.

First, the faculty member must be motivated. He must go to the mountain, the mountain will not come to him.

Second, there must be good people at the helm of the time-sharing system to help work out the inevitable problems of adapting programs for use over a terminal.

A much easier way of using computer programs in economics is to use canned programs that are available from the program library of the time-sharing system. When using this type of program, the major problem is not technical adaptation as with the large simulation programs but proper integration within a specific course. And as before, the use of this type of program, requires faculty motivation and it assumes good people at the computer; otherwise, usable programs would not be available to the terminals.

In North Carolina we have the necessary good people running a state-wide time-sharing system: North Carolina Educational Computing Service (NCECS). There are over 50 colleges served by the IBM 370/165 at the Triangle Universities' Computer Center (TUCC), which is in Durham and owned by Duke, UNC, and N. C. State. I was motivated to see what could be done with the computer from terminals in economics. Thus, the two necessary conditions to make the technically feasible a reality were met. What I did was to integrate the computer in various ways in every course that I teach in economics.

The second part of this paper details the specific programs and courses in my experience and focuses on the pedagogical problems of fitting computer programs into traditional courses.

The third section deals with technical problems, both with hardware and software, and their solution. Some tentative conclusions based on my several years experience using computer programs as a teaching tool and some optimistic plans I have for future uses of the computer will be aired in the fourth and final section of the paper.

I make no apologies for the fact I use "someone else's" programs, i.e., I did not write any of the programs I use myself. We have come to the point in the road, in my opinion, at least in the area of business and economics, where proliferation of programs often reveals a very inefficient duplication of effort. With good documentation properly disseminated, time and money can be used much more efficiently by adapting existing programs than by writing new ones. Effort can then be concentrated on the problems of implementation and the development of new teaching units employing the computer. It is the purpose of this paper to shed some light on the problems encountered in these two areas.

Pedagogical Problems and Preparations

A computer program can not be simply added to or inserted into most courses without the pre-planning similar to that necessary when audio-visual supplements such as films are incorporated in a course. First, the instructor must have a firm grasp of the substantive matter of the particular program he intends to employ. He must be certain that it is relevant to and demonstrative of the principles and concepts he is teaching in the course. Some of the simpler simulations that purport to give students a realistic learning experience actually teach theoretically incorrect approaches because over-simplification leaves out some of the variables necessary to reach correct decisions. This is the case in some of the simple general business and economics simulations in wide use. This is not to say that such simple simulations are not useful additions to some courses. For the typical freshman-level introduction to business course, they serve well to give undecided majors a taste of what a business career entails and insight into the kind of knowledge necessary for success in business. Some of these simple

general business simulations have options that allow significant increases in complexity making them much more realistic and appropriate for certain advanced undergraduate courses. On the other hand, some simulations are so complex that they can only be used for advanced graduate courses. Reality, of course, is quite complex, so that the more reality the instructor requires in a simulation the more complexity in the computer simulation he must be willing to accept. There is, then, a decision that any instructor contemplating inclusion of a simulation in a course must make whereby he trades off some reality in the simulation for some reduction in the complexity in the computer program and in the amount of capability necessary for his students to gain some advantage from the simulation.

In corporate finance and several other business and economics areas, it may be a better approach to use a complex realistic simulation almost exclusively as a second course in the subject following the first course in the principles, concepts and theoretical tools required to handle the problems posed by the simulation. (Many students have suggested this to me.) This follows from the fact that simulations in finance (and other business areas and economics generally) are primarily decision-making exercises that require the solution of certain problems. If the instructor decides that he wants to use a simulation within a single course on the subject, which is often a necessity, particularly at small liberal arts institutions where there is little chance of adding a second course, he must alter the traditional course outline in order to cover those topics where some grasp (not necessarily mastery) by the students must precede the introduction of the simulation in the course.

A major advantage of a simulation of financial management is that it requires students to deal with the whole range of financial problems right from their first encounter. This is far superior to the traditional case approach where students first learn concepts relevant to one small part of the total problem and then are asked to apply the concepts to a selected problem. Of course, the simulation approach means that students must have an exposure to the full range of concepts prior to any application. But the differences between the limited static case-method and the comprehensive dynamic simulation are worth spending more time developing theoretical material before any application of principles is attempted. I've found that I can not get the students to the point where they have a firm grasp of all the tools necessary before the point in the semester when it becomes necessary to start the simulation if there is to be enough usage of it to make set-up time and student effort worthwhile. This means that students are forced to jump in the water before they can really swim. Students take this as a challenge and very few "drown." There is a need for a book in finance (which I have written and classroom tested and is now in review) that can be used in conjunction with adequate library holdings of basic texts so students can quickly develop the necessary analytical techniques. And pushing them into the decision-making required in the simulation dramatically demonstrates to them the need to know techniques. Also, the common student criticism that a course is just sterile theory, because they can't see the applications, is obviated.

In the simulations I use, three or four students are assigned to each firm. The decisions are made either as a group or they may divide them up as they see fit. Each firm is to organize itself anyway it wants. Thus, students learn something about group dynamics and organization automatically as a by-product. Each firm is held accountable as a group for the firm's success or lack thereof; and a management report is required at the end of the simulation, both orally before the class and in writing, from each firm. The simulations count for 20-30% of the students' grade for the course, which is necessary to guarantee conscientious effort. If the simulation is not an interactive model, as some are not, students cannot alibi poor performance by citing factors beyond their control. Each firm's results are a function of only its decisions. In interactive simulations, the competitive factor is an important determinant of a firm's performance, but students rarely cite competitors' actions as reasons for their shortcomings.

I use two or three class sessions to explain the parameters, the variables and inter-relationships in the simulations. Occasionally I require the students to make a set of decisions in class. You get a better fix on who is doing what in these sessions.

After the first decisions have been submitted and run and the students have their results back, I go over their printouts with each firm privately. Sometimes I do this during the regular class time. I also tell them which firms are doing well and which are not. If I have just brought them their output, they are busy analyzing while I talk with individual firms. (It's better to have the students go to the computer center to get their output, because they then see the hardware and may develop some lasting interest in computers.) Other times I set up conferences with individual firms during office hours. The point is that when the simulations begin I cut down on lectures and begin to serve as a consultant. I get a copy of each firm's output and must keep up. If I didn't they would know it in a minute when they came for help.

Giving the firms names that are the students names is a nice touch that helps motivation. When John Jones sees the heading on his computer printout "Student, Student, and John Jones, Inc." it helps. Each student gets his own copy of his firm's printout, which is also good as students like to be seen on campus with a bunch of computer printout in hand.

This type of computer program which I've used in corporate finance and marketing teaches the students nothing about computers or programming but it is not intended to. Sometimes I give one lecture on computers and programming and the particular hardware set-up I'm using just to enhance motivation. But the computer program in this situation is merely a teaching aid. Other programs I use in other courses use the computer not only as a teaching tool but also to teach terminal operation and programming.

For example, in the first principles of economics course (macro), the students are required to use a canned XY plot routine that fits a least-squares line to their data. For data I have them use the time series of 28 macro variables listed in the cover of their textbook. The possible combinations of 28 variables taken two at a time is over 350 so that no two students are allowed to plot the same two variables. They must punch up about six cards, depending on how many sets of data they use, connect the terminal with the computer, and read in their programs. In this case they are learning how to use the keypunch, the console-typewriter terminal, and the card reader. Not too much economics is learned, but since the variables plotted are the national income accounts data this work could be inserted in the course where national income accounting is discussed. Better students used the opportunity to test some elementary theoretical propositions, e.g. the validity of a straight line consumption function and the Phillip's curve. Students who completed the computer assignment (and this was all but 5 in a class of 90) were given a "C" for just turning in the program and the output. To enhance the economics-learning aspect, I gave a "B" or an "A" for a written analysis of the output. The computer work constituted 10% of the students' grade.

None of the students had any background in computers or programming. I gave one lecture on computers and programming in general and on the particular hardware they would use. They were given three sheets of instructions, which I took another class period to go over. One was instructions on how to use the key punch; one told how to use the terminal; and the third explained, with an example, the program they were to use. There were very few problems at the terminal. My student assistant put in two hours each evening for ten days at the terminal, but problems were minimal.

In my intermediate micro theory course I use two conversational programs. The first is a teaching unit on diminishing marginal utility and it fits in right after the classical explanation for the downward sloping demand curve. Students are given an income and some consumer choices. Through choices they describe their own utility functions. The object then is to maximize total utility. The program contains four questions at the end which serve as a write-up. The other is a cobweb model of supply and demand and it is used right after the long-run equilibrium in perfect competition has been developed. I count the computer work as 10% of the grade here also, but I require a higher quality analysis of the output for a "B" or "A" grade.

In intermediate macro theory I use a computer problem kit that contains an already punched program of a basic model. It's in FORTRAN and the booklet has a chapter on FORTRAN to help the students learn some programming in order to develop the model. Preparation of the students for this computer work consists of a review of principles for which I require a programmed paperback book.

Some of the students in the intermediate level courses have had prior computer experience, but those who have not seem to do just as well. Thirty percent of the grade is based on the computer work in the macro course.

Technical Trials and Tribulations

It is essential that any instructor planning to use a simulation know enough about the technical details of the program to be able to communicate with the computer center personnel. This is very important in getting any program to run. Computer center personnel do not know, nor should they be expected to know, the substantive matter and technical details of all programs they are asked to run. Too many potential computer users know too little about computers - about what they can and cannot do and generally about how they operate. There is, then, the possibility of a rather large communications gap between potential users of the computer and computer center personnel. I suggest that the best way to obviate this problem is for the potential user to learn some programming. General introductory sessions by computer center personnel, programmed self-study courses in programming, and regular undergraduate classes in programming, are all ways of closing the communications gap and overcoming the general psychological barriers inhibiting computer use. It is not necessary to actually become a programmer. The common faculty comment that programming is just a low level technical skill not worthy of their time or effort implied by "if I need it, I'll hire a programmer" does have some validity. However, the study of programming for the potential user of the computer is about like the study of accounting for the potential businessman. He may never actually do it himself but it gives him the ability to communicate with those who do; it establishes a common language; and it leads to general insights that further close the otherwise wide communications gap. The onus

is on the faculty member to make the first move. Closing the communication's gap is the most important step in overcoming the inevitable technical problems with any program.

Five years ago I brought my first program to the computer center which had a newly installed IBM 360/20 with 16K core storage. As the operating system took up 6K, the remaining 10K was not sufficient to handle the simulation. I was presented with two alternatives: (1) convert the program from cards to punched tape and put the program on over a time-shared teletype they had hooked up to the 360/75 at TUCC, (2) send the program to TUCC for them to put on the system, call in the input by phone and get the output back by mail. I chose the latter, and by duplicating and sending TUCC a number of column coded input (decision) sheets, the arrangement worked tolerably well except for the few times when the mail was slow. What was scheduled for classes in the course became subject to sudden adjustment when the output did not come back when expected. Also, the students were not always able to have enough time between decisions if we were going to get in a sufficient number of decision-making periods during the semester.

Later, the core of the 360/20 on campus was increased to 32K and it was possible to run it there. I also used the same computer when I taught the course at a nearby college. But since that time I have used terminals for most of the programs I use. One exception is when I taught marketing in the evenings at a junior college, largely because they have a medium-speed terminal in the N.C.E.C.S. system (an IBM 2780) that I wanted to try but they preferred to segment the program and run it on a 1620 they have that's not too busy.

The first problem encountered in modifying the simulation program for an IBM 1050 terminal was the slowness of the card reader; the deck is over 800 cards. I got around this by reading in the deck over a 2780 located at another local college, also time sharing at TUCC. Another problem encountered is the fact the 1050 terminals I used did not have punched card output capability and the simulations called for punched card output. I tried to modify the program for successive runs. I was not able to do this, so I modified the program to have the punched card output printed and sent back on the 1050 printer with the output. At Computer Center student assistants would then convert this back to punched cards for insertion with the input for successive runs. However, the printout of what would have been punched cards was not column coded and it took too much time to get it punched in the right fields. I tried to work-up a program (drum control card) for the keypunch to facilitate this punching, but the keypunches I had to work with did not have left zero capability. The keypunch at the local college with the 2780 was used for this punching as it did have this feature. This arrangement is only practical when just a few firms are in the simulation for a few runs. With a normal class of 30 and eight runs, literally thousands of cards have to be punched. I looked into modification of the 1050 terminal to allow punched card output and found that it can be done for a nominal sum but the NCECS operating system did not allow punched card output to be sent to 1050 terminals. (They intended to include this capability in the system soon.) So I now run these programs over a 2770 terminal at another college, in the system that does have punched card output capability. Upgrading to the 2770 is not desirable in my case because it cannot be operated in the CPS mode, again because of NCECS operating system, (They intend to change this also) and the operating costs of a 2770 terminal are not competitive with stand-alone mini-computers nor with IBM 1130's. Some other makes of terminals are being investigated by NCECS that will provide the necessary capabilities cheaper. With modifications in the operating system most of the technical problems will be solved.

Conclusions

There is a progression in teaching techniques in this general subject matter area, at the top of which stands computer simulations. The oldest and poorest approach is the institutional, which is purely description of the institutions in any way involved or related to the particular aspect of business or economics being studied. Next comes the textbook with some end-of-the-chapter problems that give students a chance to try out basic theoretical concepts developed in the text. Then there is the combination text-and-casebook approach, with varying amounts of text and with varying length cases. Sometimes the text is long and the cases short. In that situation the case is overly simple; the student sees right through it; and it's unrealistic. When the text is short and the cases long, the student has insufficient background in concepts to do much with the case. Some books have even gone all the way and eliminated the text completely, thereby leaving the student to develop general principles as he struggles to solve the cases.

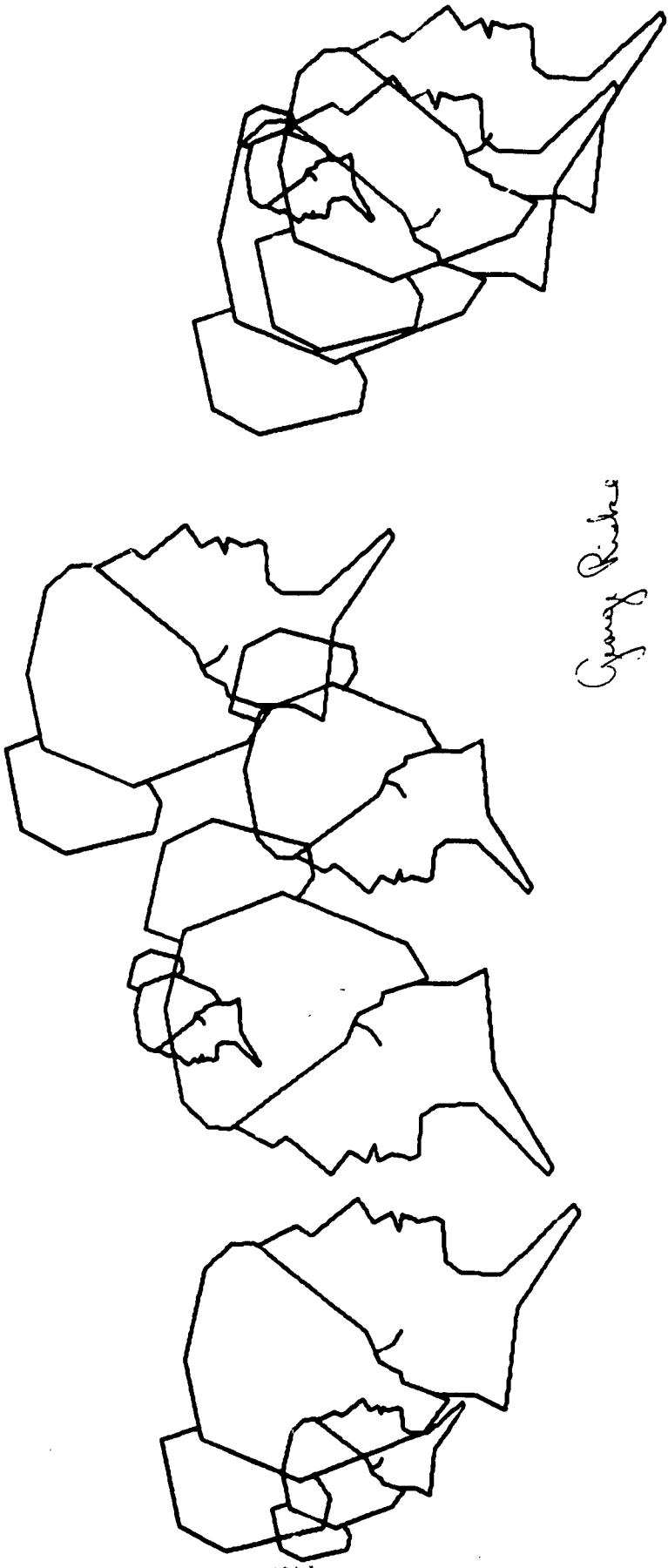
The computer simulation sits at the apogee of teaching approaches because (1) it is not a piecemeal approach like most texts, i.e., it combines all elements into a reasonable whole before the student is asked to cope with problems. This, of course, makes it somewhat more difficult for the student because he is not applying a single concept to a simple problem. He must decide what principles apply where, and this itself is a big lesson. (2) A simulation is a dynamic representation of reality. Textbook cases, no matter how complex and realistic, are static; but the output from one run of a simulation becomes important input data for the next

run. If students make mistakes they must, just as in the real world, live with them. The computer simulation represents a living dynamic case and a superior teaching environment.

The generally high level of motivation stimulated by using computer programs in economics is perhaps the most significant aspect of using them. There are a few other problems and advantages of using computer programs in economics not mentioned above, but the advantages outweigh, by so much, the problems, that I have planned the entire economics curriculum with at least one program in each course.

In addition to the six courses mentioned above, I plan to use a macroeconomic policy game with 10 identical countries in the international trade course. For money and banking, a simulation of the effects of policy decisions on money supply, investment, taxes, etc. to minimize both unemployment and changes in the price level is planned.

My experience leads me to conclude that large scale computerized simulations are no longer the exclusive province of universities with large stand-alone computers. They can be used at small schools with low-speed terminals. And, of course, smaller simulations and canned programs are handled easily via terminal. I would suggest that the minimum hardware configuration include punched card output capability, but everything mentioned in this paper except the programs calling for punched card output can be run over a teletype in a time sharing system if the instructor is motivated and the system has the kind of people we are fortunate to have at NCECS.



George Riske

WOMAN by George Riske

Problem: Continuous line design/non-maze pattern

CETERIS PARIBUS METHODOLOGY AND COMPUTERIZED
ECONOMICS-BUSINESS MODELS

Richard A. Stanford
Furman University
Greenville, South Carolina 29613
Telephone: (803) 246-3550

Although there are available on the market a wide variety of computer programs for use in economics and business courses, and the development and use of many more unpublished programs have been reported to groups of computer users, the application of the computer to economics and business classrooms is yet in its infancy. We still have hardly learned to control the computer, much less to exploit fully the teaching capacity of this amazing result of man's ingenuity.

The range of programs published or otherwise in use is extremely broad and diverse. Professor Robert S. Holbrook, in a paper presented at the 1970 Conference on Computers in the Undergraduate Curricula, put the problem in the following terms:

...The possible range is very wide, from a simple model not unlike the blackboard version already described in the theoretical portion of the course, to one of several full scale econometric models of the U. S. economy already actually used for forecasting and policy purposes.

I believe that both the very simple and the very complex models have serious deficiencies with respect to their use in a policy game context, and that a model between the extremes is most desirable. The simplest model, by its very nature, must omit most of the interesting and important problems that distinguish the real world from the simple classroom model... Full scale forecasting and policy models, by definition, incorporate real world complexity, but the enormous number of inter-relationships makes it difficult or impossible even for the instructor to trace the paths of causation from policy to goal[1].

Professor Holbrook's objections to the very simple and the highly complex models are well taken. I shall offer three additional critical observations about the variety of programs which I have seen or heard discussed:

1. Most of the programs designed for use in economics classrooms are, by model design and program input and output formats, isolated entities with little relation to other available programs. If the instructor wishes to use the computer to elaborate a variety of economic mechanisms, he is confronted with a bewildering array of different program languages, different program input and output formats, and designs for different types of computer equipment. This situation is confusing to both instructor and student, especially if they have only limited background in computer science.
2. Although some of the available programs are so simple and trivial that they accomplish little more than do the usual blackboard expositions, most of the business oriented game programs purport to cover as many of the routine aspects of the firm's operations as possible. The designers of economics and business game models often have pursued the maxim: the closer to the "real world," the better the model. The outcome of this sort of model-building behavior has been the appearance on the market of numerous hyper-complex game models.
3. While there are of course available models of complexity between the extremes, nearly all of the game models which I have seen, whether business or economics, micro or macro, seem to have a disconcerting lack of flexibility for starting simply and progressing in complexity.

I have mixed feelings about the use of computer models for classroom teaching purposes. On the one hand, I am greatly impressed with what I think is tremendous teaching potential inherent in the concept of student-participation computerized models. On the other hand, while the available simple programs accomplish little more than blackboard expositions and lead the students into a false sense of security in understanding relationships, the more complex programs which I have seen seem to have the effect of frustrating the student, rather than instructing him[2]. While the complexity is more lifelike, or "closer to the real world," the frustration resulting from the complexity seems far to outweigh the instruction which the student receives.

These results are particularly disconcerting to an economist who has been brought up on a steady diet of ceteris paribus methodology for analyzing real-world economic (including business) phenomena. Since the economist has little access to the controlled laboratory conditions of the natural scientist, he must in some other way abstract from the complexity of the real world if he is to understand and teach others about the operation of economic

mechanisms. He does this by sorting out in his mind what he thinks is the relevant from the irrelevant information. Having identified the "extraneous variables," he assumes them constant (the laboratory scientist would in fact be able to accomplish the constancy of most of his extraneous variables). The ceteris paribus analytical methodology, then, is the assumption that all variables other than the ones explicitly under study are held constant.

Having abstracted from the real-world complexity, the economist may then proceed to structure a model to exhibit the bare-bones characteristics of a real-world mechanism as he discerns it. Given the structure of the model, he may draw tentative conclusions about the operation of the model[3]. These tentative conclusions may be tested using correlation-regression analysis on real-world data which have been adjusted statistically to approximate the assumed constancy of the extraneous variables. The ceteris paribus statistical methodology, thus, is the statistical adjustment of data to approximate the assumed constancy of the extraneous variables. An example is the deflation of money value time series data by a price index to approximate price constancy over the period of study.

Having ascertained the validity of his model, the economist finally may attempt to instruct his students about the operation of the real-world mechanism by teaching the operation of the model, and relating model conclusions (implicitly or explicitly) ceteris paribus assumptions.

All of this is very well for the economist in his office, his professional journal, or his classroom. But I think that we have failed in large measure when constructing computer models for classroom use to implement satisfactorily our ceteris paribus methodology as a teaching device. Either we dwell upon the methodology by constructing near trivial models, but fail to relate them to the real world, or we completely avoid using it when we construct the hyper-complex models. An example is provided by the game models usually used in business policy courses. If the complex business game models reaching the market give any evidence at all, either the economist's ceteris paribus methodology is considered to be useless in teaching the operation of business or economics mechanisms, or the program writers have too often missed the point of the economist's procedure.

Section II

Professor Holbrook, in the same paper quoted above, expresses an opinion and makes a recommendation for alleviating many of the objections expressed above:

The best model for game use is one which is sufficiently complex to include important dynamic characteristics of the actual economy and, at the same time, is simple enough for the instructor to feel that he fully understands its behavior. To achieve this goal I believe that, if at all possible, the instructor should build his model himself, rather than use one constructed by someone else[5].

Although construction by each instructor of a model with just the right level of complexity hardly facilitates the implementation of ceteris paribus methodology as a teaching device, or solves any problems for those who have little or no background in computer science, I decided that Professor Holbrook's recommendation for self-enlightenment was worthy of pursuit. I set out to construct a set of computerized game models designed explicitly for use in a number of economic theory courses. The game models were intended to cover a wide variety of economic mechanisms, and to have similar input and output formats for ease of use by the instructor or student. Rather than design models to meet some acceptable level of complexity, I attempted to construct the models to feature options for varying degrees of complexity. The objective was to structure the models to permit progression from relatively simple contexts to almost any level of complexity desired by the instructor. The method used was to admit additional features to the models in a stepwise fashion after students had been given an opportunity to master each previously added feature.

In addition to the game models, several simulation or demonstration models were structured. These models permit the instructor to demonstrate in the classroom with remote computer terminals (or with an opaque or overhead projector to illustrate previous computer runs) the effects of changing one or as many of the variables in the model as the instructor wishes to change. Since the simulation models are designed to reflect the variables in the game models, they may be used by students to test the effects of variable changes prior to game decisions. Thus, the simulation models provide the vehicles for implementing the ceteris paribus methodology as a teaching device. The simulation models permit the student to manipulate variables in a rarified atmosphere of constant conditions; the game models provide the student with experience in manipulating the variables under conditions of interdependence approaching the complexity of the real world.

While the desire to develop adequate programs for use in microeconomic theory courses provided the initial impetus for the developmental effort, it became apparent at an early stage in planning that an extension beyond the micro theory of the firm would be useful to the

instructor who is concerned with both the micro and macro aspects of the economy. Thus, general equilibrium and macro programs have been added to the initial micro theory programs. To date, fifteen different game and simulation models have been designed, and most have been used in various economics courses at Furman University during the last two years. The programs are written in the FORTRAN IV computer language for use on an IBM 1130 computer installation with an 18K memory core capacity and disk-pack data storage. The programs may be adapted to other computer installations with relative ease, and to computers with smaller core capacities by reducing the number of firms, products, market areas, or countries for which the variables are dimensioned.

Section III

The programs are grouped into three categories: the microeconomic context, the general equilibrium context, and the macroeconomic context. Microeconomic theory generally concentrates attention on the behavior of individual units in the economy - the consumer, the producer, the resource owner, and the resource employer. The first two programs in the micro series are designed to permit instructor demonstration or student simulation of the effects of changes of any of the determinants of demand or supply in commodity or resource markets. The next three programs are designed for instructor demonstration or student simulation of the effects of variable changes, respectively, in the marketing-mix control of an imperfectly competitive firm's demand function, inventory control by the firm, and the firm's control of its production function. In any of these five programs, the instructor may use his ceteris paribus methodology to change only one variable at a time to illustrate results. Students may use the same five programs to simulate the effects of variable adjustments in advance of game decisions.

Following the five teaching or simulation models in the micro series are three game context models which are designed to be used in sequence to effect a progression from a relatively simple context to more complex situations. The first model is that of an oligopoly[6] context distributor-behavior game. This model simulates the operation of firms in a single product, wholesale or retail level oligopolistic market structure, with no manufacturing. The objective of this model is to permit the student to develop his understanding of, and skill in manipulating, the basic marketing-mix and inventory control variables under conditions of competitive oligopolistic interdependence. An option permits the instructor to limit the students to a single advertising type, or to make available to the students any of a wide range of marketing-mix variables - newspaper, magazine, radio, and television advertising; promotion type; package type and color; and product service and development expenditures. The program is designed to handle any number of firms (limited by the memory core capacity of the computer installation used), and to be played for an indefinite number of time periods, usually described as the quarter. Time lags are limited to one quarter, and data may be carried forward from each quarter to the next on punched cards, disk files, or tape files. The print-out includes decision listings; cost, revenue, and profit statements; and a balance sheet.

Two other game programs are extensions of the one described above to permit the student to integrate into manufacturing, to take an additional product line, and to extend marketing and manufacturing operations into other geographical areas. The three game models may be sequenced in the sense that time-lagged data from the distributor model for any quarter may be read as part of the input data for the manufacturing model for the next quarter; data from either the manufacturing or the distributor model may similarly be read into the multiple product and market model.

The use of the three game programs in sequence permits the instructor to introduce his students into the firm operation at the simplest level of single product distribution in a single market area with a minimum of advertising types. Once the students have been given an opportunity to operate the firm at this level, the instructor may permit them to progress as far and as rapidly as he deems desirable into marketing-mix expansion, manufacturing, other product lines, and other geographical areas (including warehousing, distribution, and manufacturing functions).

The five simulation models may be used by students in conjunction with any of the three game models to simulate the effects of variable changes in advance of game decisions. Thus, the student may reduce his complex game problem to its elements in the simulation models to "pin-down" the behavior of each of the variables separately, or several together. His problem, then, is to take the ceteris paribus knowledge gained from the simulation models, and apply it in the interdependence context of the game models. He thus has the opportunity to work with each variable separately in a rarified atmosphere, and all variables together in a context which more closely approaches the real world.

There are three general equilibrium models, one of which is purely for teaching purposes, the other two of which may in concept be sequenced with the game models of the micro context. Insertion of the general equilibrium models between the micro and macro context models permits a natural transition from the micro to the macro. This is accomplished in concept by aggregating

the firms of the micro context into sectors by types of commodities produced. The transition from the general equilibrium to the macro context is accomplished in concept by aggregating the various sectors of the general equilibrium context into the business sector of the macro context.

The teaching model of the general equilibrium context is an input-output table program which may be used by the instructor to demonstrate, or by the student to simulate, the effects of sector independence within a nation as the inputs in one or more sectors are changed. The program format includes a 10 by 10 sector input-output array.

The general equilibrium context includes two game models which differ according to the difference between interregional and international trade theory. The interregional trade model is a three region, three produce, three resource comparative advantage program. The program features interregional commodity and resource flows induced by wage and price differentials between the regions. The rational operation by students of the nondurable consumer goods, durable consumer goods, and capital goods sectors in the various regions permits demonstration of regional comparative advantage production specialization, commodity and resource price equalization across regions due to resource and commodity flows, and industry relocation across the regions.

The international trade model differs from the interregional trade model only in that students are constituted as operators of the productive capacities in countries rather than regions, and international resource flows are prohibited. Student operation of the countries in this program permits demonstration of international comparative advantage specialization, and the Stolper-Samuelson hypothesis (resource prices tend to be equilibrated across national boundaries due to international trade in commodities, even though the international movement of resources is prohibited).

The macro context includes four teaching or simulation programs, and one game program which has three complexity options. The four teaching programs are designed to fit the national income accounting, commercial banking system, Keynesian macro variable, and Harrod-Domar economic growth contexts. Each of these programs may be used in the classroom by the instructor to demonstrate the effects of *ceteris paribus* or multiple variable changes, or by the student to simulate the effects of variable changes in advance of game decisions.

The macro game program provides for student participation as operators of the consumer (resource owner), business (resource employer), banking, and government sectors of the economy. The program can handle any number of countries in operation at any one time (limited by the memory core capacity of the computer installation used), and can be played for an indefinite number of time periods. The program features complexity options which permit the instructor to introduce his students to the macro variables in a closed economy without government, and then progress to admit government to the model, and open the economies to international trade. The open-economy option features international commodity trade generated by price and income effects, autonomous capital flows generated by interest rate differentials, compensatory capital flows, automatic exchange rate devaluation if the country voids itself of foreign exchange, discretionary exchange rate revaluation, and tariff and quota limitations on imports.

Section IV

Within the last two years four of the micro teaching programs and the three micro game programs have been used in intermediate microeconomic theory courses at Furman University. One of the teaching models, the resource market program, is still in the developmental stage. The seven extant programs have been used to accompany a standard intermediate microeconomic theory text[7].

The macroeconomic game program, and two of the macro teaching programs have been used for two years in intermediate macroeconomic theory courses at Furman to accompany a standard intermediate macro theory text[8]. The national income accounting and the economic growth models are still in the developmental stage. The open-economy option of the macro game, and the international comparative advantage general equilibrium program have been used in international trade courses at Furman.

The results of program use are reported as objectively as possible below:

- a. First, the results have shown no significant quantitative gains as yet. Although no less textbook material has been covered while using the programs than previously, it is not yet possible to report being able to cover more material. There is a very important reason for this. During the last two years while the programs have been in the developmental stage, the students have been acting as the guinea pigs. They have found the "bugs" in both the economic models and the computer programs as the

programs were being used. "Debugging" in this manner necessarily slows down progress in the game contexts as rounds of decisions must be delayed temporarily while repairs are accomplished. Thus, it will not be surprising to find, when most of the "bugs" have been deleted, that some more can be accomplished in each course as the programs are administered more smoothly. Meanwhile, the programs are being thoroughly classroom tested.

- b. The positive gains which can be reported at this time are qualitative in nature. It is my judgment that my students attain a much better grasp of the text material when they use the simulation programs and participate in the game contexts than they did before the use of such programs. The evidence of this is found in the types and degree of sophistication of questions which they are led to ask as a result of model use or game participation. Use of the simulation models permits the student to "break-down" a complex problem into its elements for ceteris paribus treatment. Participation in the game contexts forces the student to pull together the information from all of his previous and current coursework in the economics and business area if he is to compete successfully. And by use of the simulation models, he can solve or gain insight into many of the game problems by himself.
- c. The extant programs seem to have gone a long way toward preventing the frustration which I have witnessed on the parts of students ensnared in the hyper-complexity of some available game models. The models described above, designed for sequential and mutually supporting use, permit a gradual approach to the complexity of the real world as the instructor begins his students at a relatively simple level of analysis, and progressively adds features to the game models.

The results reported above are nearly all non-negative. There are of course possible negative effects from the use of computerized game models in the classroom. First, the instructor and the students may become enamored with the use of the machinery, whether positive results are forthcoming or not. Second, there is always the danger that the instructor will permit his students to fall into the trap of "game-playing," rather than simulating rational real-world behavior. Third, there is the danger that the student will take with him the misconception that the complexity of the real world easily may be decomposed into its uniquely identifiable elements, as in the simulation models. Fourth, there is the possibility that the instructor will permit his students to play and become bogged down in the game, to the exclusion of text material which should have been covered.

The model designers, the computer programmers, and the writers of text material to accompany the models may try very hard to prevent all of the negative effects described above. But responsibility for preventing those negative effects ultimately rests upon the shoulders of the program administrator and instructor in the classroom. The instructor must be ever vigilant, lest he or his students fall into any of those traps.

In conclusion, I have found that the combined use of the sequential game models with the mutually supporting simulation models as described herein can provide the instructor with a highly productive set of teaching devices, provided that the instructor will take pains to prevent the negative effects mentioned above.

FOOTNOTES

1. Robert S. Holbrook, "The Use of Economic Policy Games as an Aid to Student Understanding," Proceedings of a Conference on Computers in the Undergraduate Curricula, June 16-18, 1970, (Iowa City, Iowa: Center for Conferences and Institutes, The University of Iowa, 1970), p. 5.13.
2. I have arrived at these conclusions after examining numerous available program models, and having participated in giving oral examinations to students in senior-level business policy courses in which a quite complex game model was used.
3. The procedure to this point may be described as that of the extreme apriorist, one who declines to test conclusions on grounds that they must be true in an a priori sense if logical reasoning flowed from accepted premises.
4. If the analyst tests his conclusions with real-world data, his procedure may be described as that of the logical positivist.
5. Holbrook, op. cit., p. 5.13.
6. The term "oligopoly" is used in economic analysis to describe a market structure with few enough firms so that there is a significant behavior interdependence. Each firm can

identify both the action of another firm, and the actor. Reactions can likewise be felt and identified by others.

7. The micro theory text that I have used is that by C. E. Ferguson, MACROECONOMIC THEORY (Homewood, Ill.: Richard D. Irwin, 1969).
8. My macro theory text is that by T. F. Deinberg and D. M. McDougall, MACROECONOMICS (New York: McGraw-Hill, 1968).

AN UNDERGRADUATE COMPUTER-ASSISTED INSTRUCTION COURSE IN
THE EARLY IDENTIFICATION OF HANDICAPPED CHILDREN

G. Phillip Cartwright and Carol A. Cartwright
The Pennsylvania State University
University Park, Pennsylvania 16802
Telephone: (814) 865-0471

Introduction

Under grant support from the Bureau of Education for the Handicapped, and the Bureau of Educational Personnel Development, U. S. O. E., personnel at The Pennsylvania State University have developed a computer-assisted instruction course in special education. The course called CARE (Computer Assisted Remedial Education) is a completely self-contained three-credit college-level computer-assisted instruction (CAI) course which deals with the identification of handicapping conditions in children. The purpose of CARE is to give students in preschool and primary education curricula the knowledge and skills necessary to identify children who otherwise might be educationally retarded by the age of nine or ten. The course is designed to promote clinical sensitivity on the part of the students and develop in them a diagnostic awareness and understanding of the strengths and weaknesses of handicapped and normal children. Undergraduate students who complete the 30-hour course will be able to evaluate systematically children's learning potential and to formulate appropriate educational plans for the children. Three credits in the Penn State course EEC 400: Introduction to Exceptional Children is given for successful completion of the course. The course is taught completely by computer-assisted instruction.

Need

This project seeks to improve the quality of teacher preparation in the area of special education. Intensive training in special education concepts is directed primarily toward prospective classroom teachers of elementary grades in rural schools in Pennsylvania's sparsely populated counties. A high proportion of the children in these counties come from low-income families who must depend heavily upon their local schools for long-term support and escape from poverty. The situation in Pennsylvania's Appalachian region reflects a pressing national need for special education provisions. It has been estimated that 3.75 million of the nation's six million handicapped children are not receiving the special services they need. The absolute level of this lack of service is relatively more severe in schools serving the rural population than in the urban and suburban centers. The present rates of preparation of special education personnel are not sufficient to diminish the gap between needs and delivered services. It should be obvious that an alternative, or at least an augmented approach to the provision of special services to atypical children must be undertaken. An alternative is illustrated in this project: preparation of teachers of elementary and preschool children to identify and deal effectively with conditions in children which may adversely affect their school performance.

Specialists in early childhood education and special education continually stress the need for early diagnosis of educational or behavioral deviancy, followed by early intervention with programs designed to promote cognitive and social development, in order to help handicapped and disadvantaged children get off to a good start in life. It is the contention of these specialists that the early years of a child's life are extremely important in terms of personality development and intellectual development. Unfortunately, most preschool and primary level teachers have not been trained specifically to identify children who are handicapped or who exhibit behavior which may be symptomatic of future educational difficulties.

Purpose of CARE

The purpose of the course called Computer Assisted Remedial Education (CARE) is to give educational personnel the knowledge and skills necessary to deal effectively with children who have educational problems.

The CARE course is designed to prepare prospective preschool and primary level elementary teachers and other interested persons to know the characteristics of, and be able to identify, handicapped children. Handicapped children are defined, for purposes of this project, to be those children who have atypical conditions or characteristics which have relevance for educational programming. Handicapped children include children who display deviations from normal behavior in any of the following domains: (a) cognitive, (b) affective, and (c) psychomotor.

The philosophy of the course is such that teachers are encouraged to look at children as individuals. The use of traditional categories or labels is minimal. However, certain terms and

concepts related to handicapping conditions are taught so that persons who take this course are better able to communicate with other professionals in the field.

Off-line Materials Used in CARE

When a student is interacting with the computer assisted instruction (CAI) system, he is said to be working "on-line." On-line instruction in the CARE course is dependent upon additional materials which are not controlled by, nor accessible to the computer system. These materials are called "off-line" materials; they play a large and very important role in the course.

CARE Handbook. The CARE Handbook was written especially for the CARE course. The book is 420 pages in length and contains a 350-item glossary of terms used in the course. It has two functions. First, the Handbook is a detailed summary of the course material. It may be used as a reference or refresher after a student has completed the course of instruction. Second, the Handbook contains reference material to which the student must refer when he is working on-line. The reference material consists of charts, tables, student cumulative records, examples of evaluation devices, definitions, and many other kinds of information. The Handbook also serves as a readily available notebook in which students make notes of important points.

Specimen tests. The appropriate usage of three screening tests is taught. The three tests are the Denver Developmental Screening Test, the Metropolitan Readiness Tests, and the First Grade Screening Test. These tests were designed to be used by teachers and others who have not received extensive training in testing. Each participant in the project receives sets of all three instruments. Actual test administration is simulated and problem areas pointed out. Teachers are asked to score and interpret results of the simulated administrations.

Textbook. The textbook used as a supplement to the course is:

Smith, R. M. (ed.). Teacher Diagnosis of Educational Difficulties. Columbus, Ohio: Charles E. Merrill 1969.

Objectives

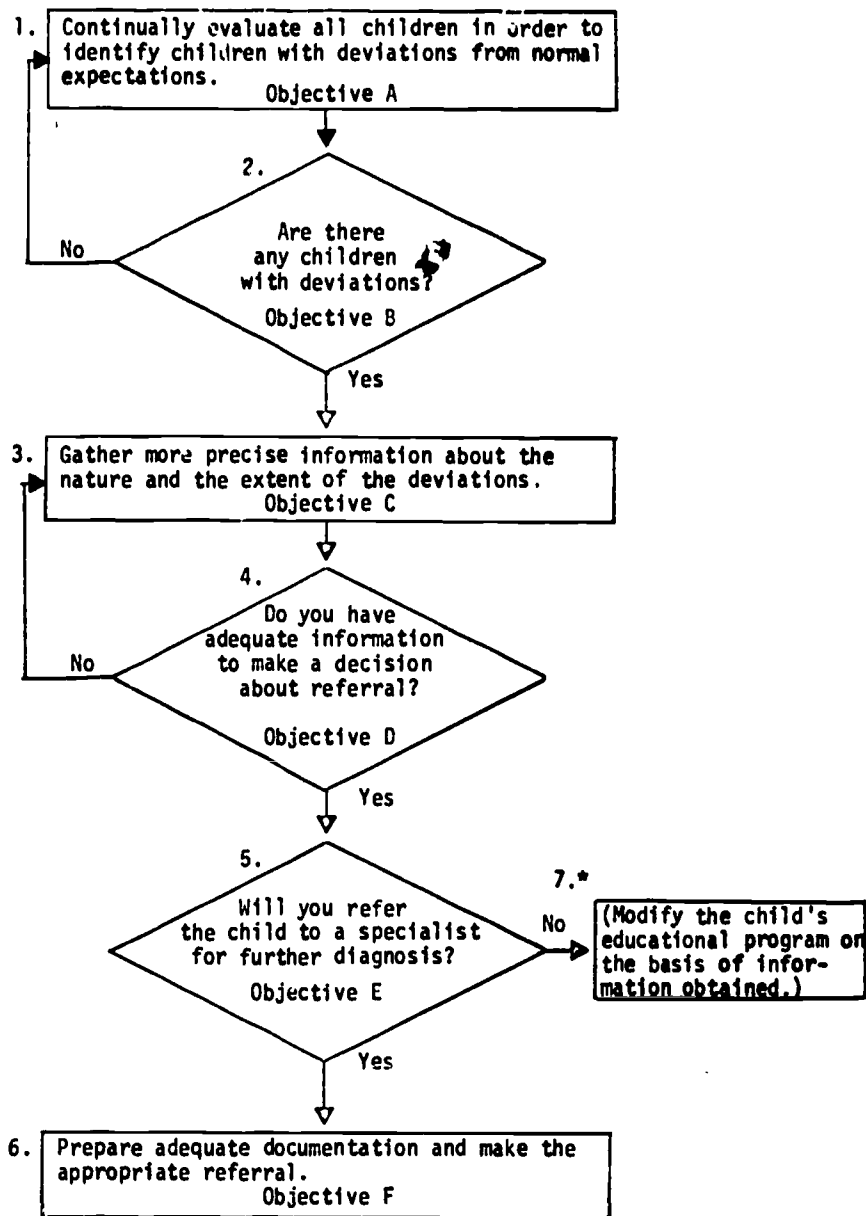
Upon completion of the CAI course, participants will have achieved the following objectives, which are directly correlated with the decision process flowchart shown in Figure 1. Participants will:

- A. know the characteristics of handicapped children and be aware of symptoms which are indicative of potential learning problems.
- B. be able to screen all children in regular classroom programs for deviations and determine the extent of the inter-individual differences.
- C. be able to select and use for those children with deviations appropriate commercial and teacher-constructed appraisal and diagnostic procedures in order to obtain more precise information as to the nature of the deviation.
- D. be able to synthesize information by preparing individual profiles of each child's strengths and weaknesses on educationally relevant variables.
- E. be able to evaluate the adequacy of the information available in order to make appropriate decisions about referral to specialists.
- F. be able to prepare adequate documentation for the case if the decision to refer is affirmative.

It is expected that the teachers who exhibit the competencies listed above will systematically evaluate children's learning potential and formulate appropriate educational plans according to the decision process outlined in the following section.

Relationship between objectives and the decision process. The six objectives are directly associated with the first six steps (boxes) in the decision process. The first two steps in the decision process dictate that the teacher evaluate all the children in the classroom in order to identify those children who exhibit deviations from normal behavior. Objectives A and B are related to the first and second steps in the decision process.

Evaluation should be thought of as a continuous process which is an integral part of the total educational process. The evaluation process includes two major tasks: (a) obtaining both quantitative (numerical) and qualitative (categorical) data about children's abilities in the cognitive, affective, and psychomotor domains, and (b) making value judgments about these data. To identify children who exhibit deviations from normal expectations is to make a value judgment that a particular behavior is considerably different from that which is displayed by a majority



*This step is the subject of a CAI course to be developed.

FIGURE 1. Decision Process

of the child's chronological age peers and is, therefore, different from the behavior usually expected of children in that age group.

In order to make appropriate educational judgments (i.e., judgments which result in educational planning aimed at intervention for the purpose of preventing potential learning problems, correcting existing learning problems, or enhancing learning assets), teachers need information about the atypical conditions and characteristics which are likely to be present, to some degree, conditions and characteristics which are likely to be present, to some degree, in groups of school age children. Information concerning both normal behavior and possible abnormal behavior in each of the domains (cognitive, affective, and psychomotor) is the prerequisite for the task of screening children in terms of deviations. It is assumed that inservice teachers possess adequate knowledge concerning normal behavior and operate, in general, with expectations of normal behavior for the children in their classrooms. The investigators maintain that the majority of inservice teachers have not had an opportunity to acquire extensive information about possible deviations, or abnormalities, in behavior which influence learning. Therefore, course content used in association with objective A provides the basic information which is the prerequisite for the screening task (steps one and two) and for subsequent tasks in the decision process.

The following items are examples of the course content for objective A: (a) definitions of atypical children, (b) descriptions of various groups of atypical children such as mentally retarded and emotionally disturbed children; (c) descriptions of children with speech, motor, auditory, and visual problems; and (d) justification for the use of certain variables in describing atypical children. Since the course is intended for teachers working with preschool and primary level children who may not yet manifest clear-cut signs of atypical behavior, teachers are given information relative to the more subtle clues to incipient problems.

Acquisition of the prerequisite information allows the teacher to identify or screen out, those children who exhibit deviations from normal behavior. Achievement of objective B enables the teacher to make correct use of data which are usually readily available to classroom teachers. Course content directed toward objective B focuses on the following: (a) the relative nature of normality in terms of socio-cultural factors, and societal and educational expectations, (b) inter- and intra-individual differences; (c) interpretation of information which is generally available for all children in the group such as results of group intelligence, readiness, and achievement tests, questionnaire responses concerning home and family, and so forth, and (d) the continuous and circular nature of the screening process.

During the first phase of the decision process, the teacher surveys the entire group of children for performance on certain relevant variables in order to select those individual children who exhibit deviations of a sufficient degree to warrant more intensive diagnosis. With the completion of the screening at any one time, the teacher will have formulated "suspicions" or hypotheses about some of the children in the group and will proceed to the third step in the decision process for these children. It should be noted that the teacher would continue to use the screening process as new group data become available.

During the third step in the decision process, the teacher gathers precise information concerning the nature and the extent of each individual child's deviation. Objective C is associated with this step. At this point, the teacher adds information about each child's intra-individual differences to that previously obtained (in the first step) about the inter-individual differences. The teacher needs to obtain data concerning discrepancies within the individual's growth pattern (the child's specific abilities and disabilities) for each of the children selected during the screening process.

Achievement of objective C enables the teacher to perform at the third stage of decision making. Course content for objective C includes: (a) rationale for use of a variety of appraisal procedures, (b) use of commercially prepared tests and non-testing materials; (c) techniques of constructing and using teacher-made tests and non-testing procedures, both formal and informal, (d) criteria for selection of appraisal procedures with emphasis on validity and reliability relative to a variety of purposes; (e) sources of information about the child from other individuals, such as peers and parents; (f) use of day-to-day informal situations, devised by the teacher, to yield information about attainment of specific behaviors of interest. The emphasis at step three of the decision process, and for objective C, is on individualizing appraisal for each child in terms of the deviations noted during screening. The teacher seeks information in addition to that which is usually available for all children, and this information will be unique to the deviation for which the child was screened out of the total group.

Tentative completion of the third stage in the decision process, together with achievement of objectives D and E, enables the teacher to evaluate the comprehensiveness of the obtained data and, therefore, make the decisions required in steps four and five. Course content associated with objective D includes: (a) description of profile charts and related diagrams, (b) procedures for selecting certain variables for inclusion in an individual's profile, (c)

interpretation of normative data; (d) rationale for the use of various kinds of information, from a variety of sources, in combination; and (e) techniques of constructing and using profile charts and related diagrams. Course content for objective E consists of: (a) criteria for determining the comprehensiveness of the obtained data; (b) information concerning the specialists who can be expected to provide various types of intensive diagnostic services for children, and (c) descriptions of the classroom teacher's role in relation to the roles of various specialists.

If the teacher makes a negative decision at step four, he needs to return to step three and collect the information required to complete the child's profile chart before proceeding through to step five. However, if the teacher is able to make an affirmative decision at step four, he will proceed immediately to the next decision block, which is step five in the process.

In formulating an answer to the question posed at step five, the teacher asks himself: Have I exhausted all sources of information available to me in my role as a classroom teacher? Can I make educational plans for this child on the basis of information currently available? Do I need more information before making educational plans for this child?

If the decision at step five is for referral, the teacher will proceed to step six. Objective F is related to step six. Course content associated with step six includes (a) criteria for selecting the appropriate specialist for various types of referrals, (b) procedures to be used in documenting the request for referral, (c) descriptions of general procedures to be followed in making referrals, (d) activities which might be required of the teacher subsequent to requesting a referral, and (e) feedback to be expected by the teacher relative to disposition of the referral.

If the decision for referral at step five is negative, the teacher will be responsible for modification of the child's educational program within the regular classroom setting (step seven in the decision process). It is not possible in this one course to deal with extensive modification of programs. A second course is planned to cover this problem. Modification of programs for atypical children would include the following topics: (a) techniques of effective classroom management, (b) specialized teaching strategies which might be used for amelioration of difficulties, or for enrichment, in various subject-matter areas, (c) special materials to be used in association with specific strategies, (d) sources of information regarding specialized strategies and materials, and (e) resource persons usually available to assist classroom teachers.

Computer-Assisted Instruction

Instruction is individualized for the teachers by means of computer-assisted instruction (CAI). CAI presents instruction in an environment where the material presented to the learner is selected and sequenced, with the aid of a computer, to be responsive to the individual learner's needs. The computer selects sequences of instruction which are appropriate to an individual's background knowledge of the course content, his rate of progress through the material, and the types of errors (or non-errors!) the student makes as he interacts with the system.

Because each student can communicate with the system independently, and since the computer can arrive at logical decisions based on its analysis of incoming student performance data, the capability exists for the intelligent adaption of instruction for each student. The logical decision-making ability of the computer, along with its extremely rapid access to large volumes of stored information, combined with the knowledge and skill of the author-programmer, can provide for a wide variety of individual differences among learners.

To accomplish the course objectives outlined above, an 8-terminal IBM 1500 Instructional System has been installed in the Computer Assisted Instruction Laboratory at Penn State. The IBM 1500 Instructional System consists of eight instructional stations each with cathode-ray tube display, light pen, typewriter keyboard, audio device, and image projector. The computer equipment is comprised of an 1131 central processing unit, 1442 card reader and punch, 1133 multiplexer control unit, two 2310 disk storage drives, 1502 station control, two 1518 typewriters, and two 2415 tape drives.

The central processor of the IBM 1500 Instructional System is an IBM 1130 computer with 32,768 sixteen bit words of core storage. In addition to the usual peripheral equipment, the central processor depends upon three IBM 2310 disk drives (1,436,000 words) for the storage of usable course information and operating instructions. Twin magnetic tape drives record the interaction between the program and the student for later analysis and course revision. Core storage cycle time is 3.6 microseconds and read/write time for disk storage is 27.8 microseconds per word.

Each IBM 1500 student station consists of four optional display/response devices which may be used individually or in combination. The central instrument connected to the computer

consists of a cathode-ray tube screen with sixteen horizontal rows and forty vertical columns for a total of 640 display positions. Information sufficient to fill the screen is available in micro-seconds from an internal random access disk. A light pen device enables the learner to respond to displayed letters, figures, and graphics by touching the appropriate place on the screen. A part of the CRT device is the typewriter-like keyboard which makes it possible for the learner to construct responses, have them displayed at any author-desired point on the CRT screen, and receive rapid feedback in the form of an evaluative message. Four dictionaries of 128 characters each of the course author's own design can be used simultaneously. An image projector loaded with a 16mm microfilm is capable of holding 1000 images on a single roll and of accessing forty images per second under program control. An audio play/record device based on four channels on a 1/4-inch tape is an integral part of the system. An electric typewriter on the system is a separate device which enables the proctor to receive a paper copy of information about students and course progress. Figure 2 shows the system configuration. Figure 3 is a photograph of a student working at the CAI terminal.

Evaluation

Formative evaluation. Several course revision cycles involving 400 students were used in the formative evaluation phase of this project. Student performance records were obtained by means of an automatic recording system. The system records data such as exact responses for each frame and course segment, number of requests for help, time of response, response latency, number of attempts made by each student to each question, correctness or incorrectness of response, type of response required, contents of displays and registers, and other similar data. Data of this nature are invaluable in diagnosing errors in content or logic and in evaluating the extent to which students are reaching each of the 1100 sub-objectives of the course.

A total of 333,018 separate student responses were analyzed and extensive revisions were made on the basis of the analysis. Each of the revision cycles resulted in noticeable improvements in student performances.

Summative evaluation. During the Winter Term, 1971, a summative evaluation of the CARE 1 program was made. All students who were enrolled in EEC 400: Introduction to the Education of Exceptional Children were randomly assigned to either of two conditions - Computer-Assisted Instruction (CAI) or Conventional Instruction (CI). The CAI group (n = 27) received all instruction by means of the IBM 1500 Instructional System and did not attend classes with the CI group. The CI group (n = 87) received the conventional lecture-discussion method of instruction and met three days per week in 75 minute sessions for ten weeks.

All students, CAI and CI, were enrolled as regular students for three credits of undergraduate or graduate credit. Both the CAI and the CI courses were designed to reach the same objectives. The instructor of the CI group was an author of the CAI course and helped plan the structure and the objectives of the CAI course.

The dependent variables in this investigation were time and final examination scores based on 75 items. Results are shown in Table 1.

	Final Examination Scores		
	\bar{X}	S.D.	t
Computer-Assisted Instruction	65.69	4.68	
Conventional Instruction	52.78	5.89	11.65*

*This difference is statistically significant with $p < .001$.

<u>Time</u>	
Computer-Assisted Instruction	\bar{X} = 25.21 hours per student
Conventional Instruction	37.5 scheduled hours per student

TABLE 1. Results of Summative Evaluation

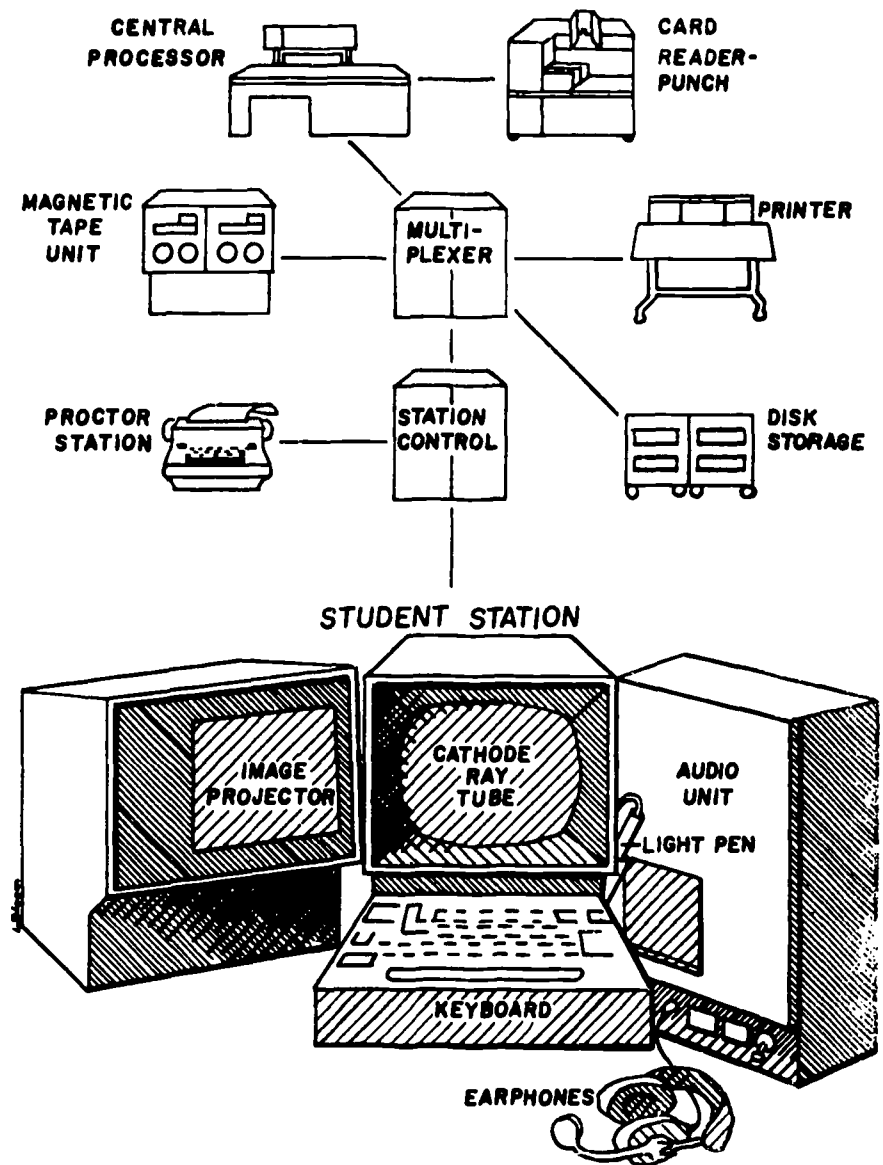


FIGURE 2. Configuration of the IBM 1500 Instructional System.



FIGURE 3. Student working at a CAI terminal.

185

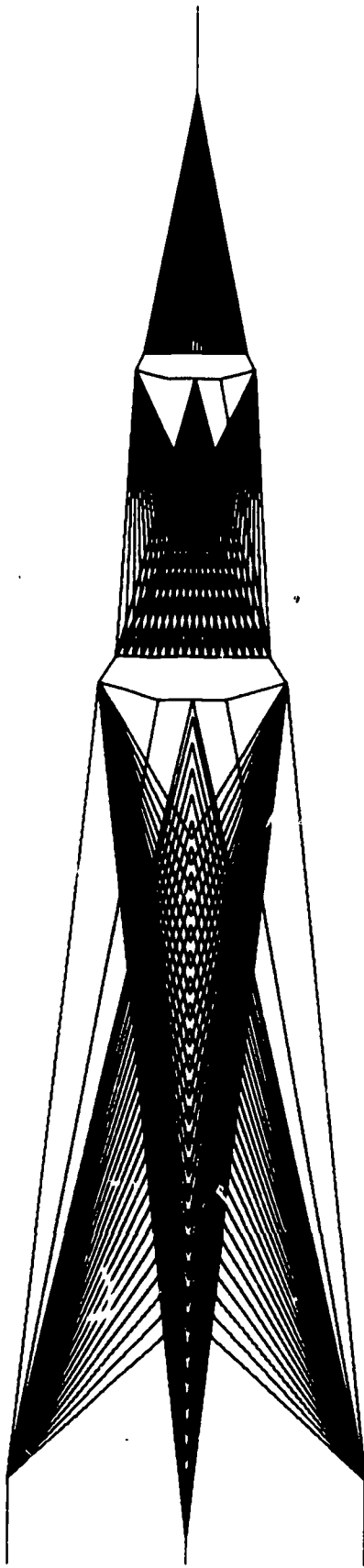
These data indicate that the group of students instructed by CAI obtained a mean score 24% higher on the final examination than students instructed in the conventional manner. Furthermore, the CAI students completed the three-credit course in 12 hours less time (33%) than the conventionally instructed students.

Modes of Instruction by CAI

The CARE course uses a wide variety of instructional strategies to assist students in reaching the course objectives. All the strategies are interactive and all require active involvement on the part of the learner. The most prevalent strategy used in the course is the tutorial approach. This approach simulates the master tutor engaging in an interactive dialog with an individual student. The tutor presents information, asks penetrating questions, and carefully analyzes the student's responses to the questions. On the basis of the student's demonstrated understanding or lack of understanding of a given concept, the tutor provides alternative courses of instruction, remedial sequences of instruction, or even enrichment material. The tutor can move a capable or well-informed student through a course of instruction very rapidly. Similarly, the tutor can tailor a sequence of instruction to meet the needs of a student who is not as capable or does not have a good background of experiences or preparation. The sophisticated CAI system can perform the chores of dozens of tutors rapidly and efficiently. The net effect is that hundreds of teachers in the CARE project have been individually tutored in certain special education skills.

The second major mode of instruction used in the CARE course is the inquiry approach. This type of activity is used in the latter stages of the course to draw together all the concepts acquired by the students throughout the course. This strategy includes simulation of regular classroom problems as well. In essence, the inquiry and simulation approaches as used in the CARE course are directed problem solving strategies. Students are told that they have access to information about a class of first-grade children. One or more of the children in the class may be handicapped or have an educational problem of one kind or another. It is the student's task, in effect, to screen the class for children with educational problems, identify those children with potential or existing problems, and deal with the problem by modifying the child's educational program or making an appropriate referral. The student begins the screening by looking over the complete cumulative records of the children in the class. The student may ask the computer for additional information. Not all the information the student receives is accurate; in fact, many false leads are given to lure the unwary student into making the wrong decision. The computer system will lead a student down the wrong path for awhile and then explain why that particular line of reasoning is not appropriate for that specific child. Eventually, as a result of skillful questioning on the part of the CAI system coupled with the appropriate line of questioning by the student a decision is reached by the student to refer a child or to modify the child's program. The student's decision is evaluated by the system and then the student's plan for referral or program modification is evaluated by the CAI system.

When a student completes the course, he has actually constructed several case histories of children with problems and has made educational decisions related to the best plans for dealing with these problems.



ROCKET by Sanford Ma
Problem: Imaginative flight form

187

176

PREPARING MATHEMATICS TEACHERS TO USE
THE COMPUTER IN SECONDARY SCHOOLS

Roger H. Geeslin
University of Louisville
Louisville, Kentucky 40208
Telephone: (502) 636-4584

ABSTRACT

After a brief introduction which includes some background material, the paper deals with three major areas: the description of a course to prepare teachers, an indication of what to expect when the computer is introduced into the schools, and one approach to solving the problem of high computing costs. Finally, there are some general concluding remarks.

Introduction

This paper is based on experience gained over the past two years in an experimental program developed at the University of Louisville in cooperation with the Louisville and Jefferson County Public Schools. This program will be expanded and strengthened through an NSF sponsored Cooperative College School Science program which will begin on July 3, 1972, with a six-week summer institute. The course described here was also offered in the Spring term, 1972, so that any significant differences between this and a concentrated summer course may be noted in the talk. Discussion of the computer network in use will be postponed to the portion dealing with this and other practical matters.

There is something quite unusual about using the computer in any kind of teaching, and it is for this reason I have avoided the word "training" in the title of the paper. Teachers can be trained to use the overhead, movie and slide projectors. They can learn to use effectively all kinds of audio-visual aids, and I would interject here that secondary teachers seem to do a much better job at this than their counterparts in the universities and colleges. But the computer is not just another teaching aid, to be used by the better teachers and ignored by the rest. I hope to clarify the importance of a special effort to prepare teachers to use the computer effectively.

Let us look briefly at the computer, which in most time-sharing systems is present in the form of a teletype or other terminal device. Since all who attend this conference are, or should be, familiar with the distinct advantage of an interactive time-sharing environment over batch processing for beginning instructional purposes, I will not present that discussion here. However, in this paper I shall normally use the word computer to mean a time-sharing computer terminal which has available at least one easily learned programming language, such as one of the extensions of BASIC. What is the nature of the beast, other than being a somewhat noisy but speedy communication device which will become quieter and faster as technology and funds permit?

The versatility and power of the computer puts it in a class by itself as a teaching aid. It can give individual remedial drill to a pupil without the attention of the teacher. The pupil can himself make up problems and then "check" the computer (more on this later). The average student can use it to supplement his course work, giving clearer insight into concepts to be learned and allowing him to solve problems more realistic and complicated than otherwise would be possible. The advanced pupil can explore, create, and experiment in the many fascinating areas of mathematics which lie beyond the normal and advanced secondary courses. Note carefully, this single piece of equipment can catch the attention of the slowest learners and give them some success in a subject they have failed for as long as they can remember. And the next moment it provides the means of carrying out long involved computations over elaborate logical paths in some "new" area, yet unexplored by student or teacher.

These remarks should be enough to hint at the potential of the computer in mathematics teaching. Let us see now how we go about preparing the teacher for this adventure.

The Introductory Course

Various aspects of a three semester hour course open to both senior college undergraduates and graduate students will be described here in general terms as a frame of reference. The details will vary from offering to offering since such depends on the interests of the class.

The course description is as follows:

Math. 595 The Computer in Mathematics Teaching (3)
Prerequisite: One year of calculus
(no prior computer experience needed).
Introductory programming and the use
of the computer in teaching a wide range
of topics in secondary school
mathematics. Summer session.

Note that the prerequisite should not exclude any secondary school mathematics teacher, and reassurance is given to those having no experience in computing. It is very helpful if the teachers taking the course have actually taught, if only in a student teaching experience, but when the course is taken by undergraduates this may not be possible.

The course has three main objectives: to develop programming skill adequate for meaningful use, to gain significant experience in problem solving and other areas of interest to each teacher, and to require each teacher to give careful consideration to the way he will introduce the use of the computer into his own teaching in a particular school situation.

The first goal can be easily accomplished by beginning BASIC with hands-on experience the first or second period at the latest. Adding PRINT statements to existing programs encourages providing meaningful output from the run of a program, and this carries over nicely to the programs written independently later. At the beginning, most teachers have no idea how they would use the computer, but usually after four or five classes, they begin to want to try something on their own. Encouraging this interest while answering questions in class and laboratory makes for a very stimulating course. Back-up suggestions are helpful when interest lags, but these are not as important as keeping in touch with what the class is doing and sharing items of general interest.

Collecting and reading programs regularly with no fixed number required encourages the teachers to do as much as they can, and provides opportunity to suggest improvements which they can incorporate immediately. Adequate, reliable computing facilities are essential, the more terminals the better, and hours of availability should be as long as possible. Some teachers arrived as early as 7 a.m. and others stayed as late as 9 p.m. in order to have longer, uninterrupted time on the computer. However, normal constraints may encourage the preparation of paper tapes to make more efficient use of the on-line terminals. Indeed, if such will be the case in their own schools, this experience is absolutely necessary. Appreciation for scheduling difficulties and other practical problems is gained directly and indirectly through this experience.

Perhaps the most important aspect of the course is the personal experience gained by each teacher in programming independently. This is the primary opportunity to develop skill in this area since the students will tend to dominate their school terminal, and the teacher will have little time to do his own programming after introducing it in his school. So it is vital that each teacher develop his own ability as much as possible in this brief course, and it is surprising how competent many become. This experience also brings out two other facts which should be recognized. The first is that although ability varies, almost anyone can program the computer and obtain significant results. The second is that they learn more from each other and actual experience than they learn in the classroom. Both of these should be kept in mind in developing and evaluating any program.

New topics in programming should be introduced as earlier ones are mastered. As the teachers become more competent, and interest in their own programs increases, it is time for them to start organizing their thoughts around their own school situation. For this purpose a required paper provides an excellent vehicle. It may happen that their situation changes so drastically that almost nothing they planned is carried out, or they may not have computer facilities at all. Nevertheless, the careful thought given to this question is never wasted. Here again the range of approaches varies greatly, with none being the right or best way. Each teacher should find his own approach. Of course, sharing these ideas near the end of the course may avert some major blunders, but the experience gained in the course gives a very valuable guide in adapting to changing conditions. When compared with those having only prior experience in a programming course, teachers who have completed this course appear to have a distinct advantage because of the relevance of the experience gained to their own teaching situation.

Comments and evaluation of the course by the teachers has been very enthusiastic, and this enthusiasm has been demonstrated not only by the long hours spent in the course, but by the hard work and long hours given by them in their schools. It is a demanding but highly rewarding task; students get excited about their mathematics classes!!

What to Expect

When the computer is introduced into the school, many things can happen. I will indicate a few effects which might occur in the classroom and possible side effects on other students, teachers, parents and administrators.

One would expect that the computer might best be introduced to advanced classes in senior high school. This is not necessarily the case, and some of the more dramatic results have been obtained in the basic and general mathematics courses as early as the seventh grade. This does not exclude use at the upper levels, but indicates positive results at all levels of secondary school mathematics. It is not possible to document here the many cases which support the remarks that I will make concerning the various courses and grade levels, but they are based on actual experience.

In the seventh and eighth grades, motivation can be held high with very simple programming tasks. Typing ability is not well developed, but the pupils can demonstrate great patience, both in typing programs, and in developing designs to be printed by the terminal. The random number function is also popular in the development of games and simulations, giving an intuitive introduction to probability theory and opening the door to many experimental uses of the terminal. A simple coin toss program can give answers in seconds to a 100, 1000, or even 5000 toss simulation. The main point here is that a very modest introduction to BASIC and a few functions can go a long way for this level average or above average student.

Deserving of special notice is the response by students in the ninth and tenth grade general or basic mathematics classes. Here one confronts pupils who have failed to learn the fundamental operations of arithmetic, and expect to fail in such that they do in any mathematics course they encounter. It is first of all surprising to them that a teacher would allow them to use the computer, especially when some other advanced classes cannot. They quickly respond to this opportunity to have a "status symbol" and pull themselves up from the bottom of the heap. The class becomes something special and begins to do its regular work in order to get on the computer. Some prefer not to write programs when given the choice of using the terminal after their regular work is done or just doing the regular work. But notice that both alternatives include doing the regular work.

Another specific use in general mathematics is to give remedial drill to those who have not learned multiplication, for example. Such a drill can be randomized and personalized with a grade given at the end, and still be very modest in size. Several have been effectively used on our network and are available to all from our systems library. But even without the drill program, it was noted in one general math class that the pupils would program a simple calculation, and then "check" the computer, which to their surprise seemed to be right always. Here we have the pupil making up his own drill, and working toward the correct answer which before had always eluded him. The pupil can do either the drill, or this type of calculation with little or no attention from the teacher. In basic or general mathematics classes where the goals are most modest, some remarkable changes in attitude and motivation can be found as well as the learning of skills which seemed impossible.

Topics in the upper level courses of algebra, geometry, trigonometry, probability, plane and solid analytic geometry, and calculus which lend themselves to treatment on the computer are legion. These have been and are being developed and documented in many publications. But a few general comments should be made.

First, the students in these upper level courses may resent the intrusion of the computer on the class time to be devoted to a specific amount of material, and indeed this is a real problem to be faced by the teacher. The answer may be to have a separate computer course for senior high students, but such a course for a full semester may easily take the students beyond the range of the teacher's computing experience. It is interesting to find that you cannot go very far in a computing course without running into the need to know more mathematics: mathematics becomes useful and a part of the real world.

Second, the upper level students are not satisfied with learning the fundamentals of computing before going on to more complex problems. They are inclined to jump directly into some elaborate, difficult, or even impossible task and then come to the teacher for help. The teacher can hopefully convince them of the need to build a more solid foundation; a lesson which could also carry over to other aspects of their lives in preparation for further education. Coupled with the fact that some students devote themselves almost entirely to writing programs (i.e. become "computer bugs") at the expense of their other studies, these are some of the difficulties encountered at the upper levels.

But the benefits seem to outweigh by far the detrimental aspects in the classroom. One major gain in all grades (7-12) is the development of skill in logical thinking and in the analysis of problems. Experience indicates a marked advantage of those with programming experience in their ability to construct proofs in geometry or to prove trigonometric identities. The programs written were not necessarily related to the course, but the discipline of programming itself develops this skill. Students will often go to much greater lengths to

satisfy the demands of the computer language for careful logical reasoning than they ever would to satisfy a teacher.

The time-sharing computer terminal in the classroom can be a very strong stimulus to high motivation in both students and teachers. This motivation, which can carry over to mathematics and other subject matter areas, is part of the driving force behind much of the activity in this field today. There is real power to solve problems, there is wide versatility to adapt to new situations, and there is great satisfaction in the successful run of every new program.

What about the side-effects on those not directly involved with the computer? Other students want to know how they can get to use it, or if they used it last year, why they can't use it this year; perhaps a computer club would help here. Some mathematics teachers may feel threatened. These are just now getting comfortable with the "new math" and along comes the computer. Don't expect much support or interest from those not willing to take the preparatory course.

Parents also react to the computer in the school. But they are proud of the fact that their children are so interested in anything having to do with studies and have demonstrated a skill which is new and unusual. The response can be even more positive when parents find that they can also learn. In addition, the equipment for a mathematics or computing lab can take its place beside uniforms for band or athletic team, bleachers for the stadium, and other improvements to the school worthy of PTA financial support.

School administrators seem to fall into two classes: those that look on the computer as another attack on their meager budgets and those who somehow are convinced of the powerful potential of the computer in the schools. The first give little or no support and thus see little or no success. They are convinced that they are right. The second again must be separated into two groups. The one may invest large amounts in computer hardware with little or no support for trained personnel. The result is what is now a familiar disillusionment with the computer. The final group recognizes the need for balance between equipment and trained personnel; not that large sums are necessary to get started, but that sufficient funds be used well. For these, there is pride in the real improvement in their educational program, not just a gimmick or empty status symbol. And they find that their investment pays handsome dividends.

How to Get Started

The cost of computing power is decreasing and special purpose mini-computers are usually very efficient in doing the tasks they were designed to perform. If schools are willing to cooperate with a nearby college or university and all have some funds to commit to the program, then the only missing ingredient to a successful beginning is the faculty member(s) to put it all together.

The key factor in keeping the costs really low is that in a time-sharing environment one teletype can be used four days off-line preparing paper tapes for one day on-line, and this proposition seems to be a rather good one. Using this approach, five schools can share a single port and all make a significant beginning. If commercial vendors would permit this sharing of a port, then the operation and maintenance of the system could be left to them. But normally, they provide more computing power than is really necessary, e.g. a choice of languages, and do not permit such sharing.

What will cut the cost further is to have enough schools in a pool to support a small-sized computer which was designed specifically for time-sharing in a single, simple yet powerful language. The system in use at the University of Louisville is the Hewlett-Packard 2000-A with 16-terminal capacity and a very adequate, extended BASIC. Such a system is essentially a turn-key operation with a very minimum of operator intervention, and although it is not now the newest thing on the market, it is very satisfactory and reliable.

Taking such a system as an example, the monthly lease-purchase charge is under \$2,000. If ports are priced between \$500 and \$600 per month, a minimum subscription of the equivalent of four ports can pay for the system to support those ports. The cost per school for computing time with five schools sharing each port could then be between \$100 and \$120 per month. As the computer becomes available, rates can be established for users within the university to insure adequate support for the system, so that it can be expanded if that should be necessary, or fully supported by the university if the funds from the schools grew to the point where they could support their own systems.

What is outlined here is a way in which cooperation between various institutions can make possible at a modest cost to all, a beginning which none could afford independently. Even this suggestion must be adapted to fit any specific situation, and the follow-up years would probably show even more variation. I will be prepared to present further specific details at the

conference to indicate actual costs related to our situation, if there is interest in this information.

In developing this program, extensive use was made of the experience gained and materials developed by the three-year Dartmouth Secondary School Project (NSF Grant GW-2246). The major difference in detail lies in the preparation of programs on paper tape which permits the reduction of on-line time from 20 minute blocks to 5-10 minute blocks for each student. This certainly encourages the efficient use of computer time without losing the "hands-on" experience in running programs. It also gives the student a permanent copy of his program on paper tape which can be used again on any terminal having access to the same language compiler. TIES, in St. Paul, has also used an optical marked card reader to increase throughput from their teletype terminals. The chief advantage here is that the student can prepare his card deck anywhere with only a pencil and a deck of cards. We are also experimenting with this device.

Conclusion

Explicit mention has not been made of the demands on the teachers who become involved in the use of the computer in the schools. As an extreme example, the on-line day at one school tends to start at 7:15 a.m. and end somewhere around 9:00 p.m., with a group of parents wanting to have instruction after that. You can imagine all the questions this generates for the off-line days! Yet the teacher most involved at this school will give all she can to her students, who are learning like they have never learned before. There are many fine, dedicated, and talented teachers in our schools. We can and should seek them out and prepare them, along with new teachers, for an extremely demanding yet rewarding adventure, the use of the computer in teaching secondary school mathematics.

COMPUTER-MANAGED INSTRUCTION:
A BEGINNING AND A REALITY AT
WESTERN WASHINGTON STATE COLLEGE

Raymond F. Latta and Joan Straughan
Western Washington State College
Bellingham, Washington 98225
Telephone: (206) 676-3360

Introduction

The purpose of this paper is to discuss the use of the computer in undergraduate education within the Department of Education. Before beginning, however, we would like to make a distinction between two uses of the computer in education. Computer Managed Instruction (CMI) makes use of batch processing or remote terminals (on a time-sharing basis) to monitor and manage student progress through frequent testing and analysis related to specific objectives. Computer Assisted Instruction (CAI), on the other hand, is geared to an interactive terminal system only, in providing instructional sequences with specific objectives for students. Methods of instruction might be tutorial or socratic, drill and practice, simulation and gaming, or problem-solving and information retrieval.

The particular use of the computer pursued by the Department of Education at Western Washington State College, and described herein, tends to focus more on CMI than CAI. The rationale for this emphasis in the course unit being described is simply one of logic and expense. To be more specific: (1) CMI is not as costly as CAI. The hardware and software costs related to CAI tend to be higher than those related to CMI. (2) Recent emphasis on individualized instruction has necessitated that educators develop technology to assist in management and information handling. (3) CMI is less time consuming on the part of the innovator and requires less expertise, in the initial stages, than does CAI. Preparation, formative and summative evaluation of CAI materials often involves a long, tedious schedule. (4) This college has seven teacher education programs which are competency based, and new management systems are needed to maintain them.

The course which utilizes the computer within its curriculum for both presentation and course management is Education 444: Instructional Management Systems. The course, which is worth three credit hours, was developed and tested by the authors of this paper during the spring and summer quarters of 1971. In all, thirty-two students, mostly seniors, have participated in the course development and testing.

Course Content, Student Activities, and Scheduling. At first glance, Ed 444 appears to be simply a survey course dealing with manual and automated instructional systems. The strength of this course, however, evolves from the required reading, on-site visits, or working through CAI programs unrelated to the course at hand. Rather, CAI is utilized to present some of the course material, and CMI, to manage student progress through a unit of the course on a systems approach to the development of instructional materials. Students then apply the process to the construction of a small learning package/module.

The schedule for actual course content and related activities (summer quarter) is shown in Figure 1. This schedule is similar to the one used for the preceding spring quarter when eight students worked through the first trial run of the course. Emphasis on the spring session was related more to course development, evaluation of instructional materials utilized with the course, and sequencing of activities, rather than to evaluation of the course overall. At the conclusion of spring quarter, learning packages were modified on a priority basis.

As is shown in Figure 1, the student in Ed 444 is first given instruction on COURSEWRITER III Author Language, then given instruction in the development of learning packages, and finally is presented with a survey of nationally recognized individualized instructional systems. Figure 1 shows each of these units with the mode of instructional presentation used, student activities, and student products related to each unit.

As was mentioned earlier, the course tends to emphasize or focus more on CMI than on CAI. In view of this approach, it is essential to emphasize CMI through student activities and class discussion.

As illustrated in Figure II, learning packages and the computer are utilized within the instructional process for teaching the second unit. The learning packages relate to the steps in the model of the systems approach taught in this unit. The steps in the model are: problem identification; task analysis; entry behavior/readiness; performance summative evaluation; and implementation of monitoring[1]. The packages utilize a presentation format similar to that which the students are expected to follow in the development of their own packages. The primary

Activity	In Class Activities						Nationally Known Individualized Instructional Systems			
	Introduction (1 hr.)	CW III (3 hrs.)	CW III (2 hrs.)	Learning Packages (4 hrs.)	Learning Packages (4 hrs.)	IPI ² System	PLAN ³ System	IGE ⁴ System Duluth, Minn. System	Florida State Univ. System REFLECT ⁵ System	
Computer Activities Outside of Class	Program ¹ calc (1-2 hrs. student time)	Practice Authoring CW III (variable time)	Program ¹ System (1-2 hrs. student time)	Program ¹ System (1-2 hrs. student time)	Programs ¹ add and calc (1/2 hr. student time)	Students author the test questions in their learning package for use in a mini CMI system (variable time)				
Computer Use (CAI or CMI)	CAI		CMI	CMI	CAI	CMI				
	0	1	2	3	4	5	6	7	8	9

TIME (Weeks)

¹Programs (software) are discussed later in this paper under the section headed software.

²IPI (Individually Prescribed Instruction)

³PLAN (Program for Learning in Accordance with Needs)

⁴IGE (Individually Guided Education)

⁵REFLECT (Research into the Feasibility of Learnings Employing Computer Technology)

FIGURE 1. Course Content and Schedule for Computer-Related Activities for Ed 444, Summer, 1971

difference between the packages used in the course and those developed by the students is that the test questions relating to the specific objectives in the course packages have been programmed into the CMI system.

The student, in working through this section of the course: (1) works through the first package; (2) when finished with the learning package and feeling that he has mastered each objective, he proceeds to the terminal room; (3) he signs on and receives the criterion-referenced test for the particular package; and (4) upon completion of the test, he follows the directions provided by the CMI system. (The student usually proceeds to the next package and follows the above sequence of steps until he has finished this section of Ed 444.) It is this system which provides the student with base knowledge of a systems approach to be followed in the construction of learning packages. The instruction is managed by the WWSU IBM 360/Model 40 computer through on-line use of the terminal systems, and constitutes a working application of a semi-CMI system. Concurrent with and following execution of the CMI sequence, the student, in a laboratory-workshop setting, applies these skills to the actual construction of his or her own learning package. The subject matter and details of the learning package are left entirely up to each student.

The synthesis activity for Ed 444, as shown in Figure II, involves the threading together of those skills learned in the course. The activity referred to is the application and utilization of CW III, systems theory, student-developed learning package, and insight gained from exposure to other instructional management systems to construct a student-designed mini-CMI system (in a very unpolished form).

Software. The computer software used for illustrative purposes in Ed 444 would fall short of being described as very good examples of either CAI or CMI programs, nor are any claims of excellence made to the students. If a detailed description of computer software used for illustration of CAI and CMI were to be provided, it would be lacking in superlatives. No claims of excellence were made to the students. Although the shortcomings were not purposefully inserted, they contributed to the course by giving students the pleasure of being able to critique the programs in light of their new-found knowledge, and to make suggestions for improvements. As such, use of CAI and CMI programs which are of perhaps average quality should not be outlawed and may well add to the student's opportunity for learning. The following is a list and brief description of the software utilized in Ed 444:

1. cai This program using a tutorial mode, teaches the student uses of many fundamental op codes in COURSEWRITER III. The culminating activity of this short program, which takes from one to two hours to complete, is a very brief student-written course.
2. calc This short program also utilizes a tutorial mode. The program, which teaches the student how to use the computer as a desk calculator, takes fifteen to twenty minutes to complete.
3. add This course, which takes from five to ten minutes to complete, teaches a very short segment of first grade arithmetic utilizing the tutorial mode.
4. system As has been explained in the preceding pages, this unit contains the criterion-referenced tests for each of the steps involved in applying a systems approach to the development of instructional materials. It requires approximately one and one-half hours on-line terminal time.

The terminal laboratory at WWSU. All CAI and CMI on-line activities take place in WWSU's terminal laboratory. It is here that eight IBM 2741 typewriter terminals are located for student use (Figure III). The IBM 360/Model 40 computer at WWSU, with the core allocated to the terminal system, can support twenty-two terminals. Terminals beyond eight in number are usually located in individual departments. During the pilot study in the summer of 1971, the terminal system was available only two hours daily, Monday through Friday. The number of terminals and availability of the system led to some major frustrations.

Findings. Education 444, summer quarter, 1971, was evaluated both formally and informally. The formal evaluation was undertaken by the Testing Center at WWSU, and the informal evaluation, by the authors.

The formal evaluation by the Testing Center found the course to be about average in comparison with other courses on campus. Several factors were critical: (1) At the beginning of the course, twenty-one terminals were available to the twenty-four students enrolled. This number was cut to eight halfway through the quarter as the result of a College financial crisis; and (2) the terminal system was available only two hours daily, five days per week. The reader must surely be aware of the ramifications of limited terminal availability both in regard to course evaluation and the scheduling of students on terminals. The cutback on terminals also

UNIT	MODE OF INSTRUCTIONAL PRESENTATION UTILIZED	STUDENT ACTIVITIES	STUDENT PRODUCT
1. COURSEWRITER III	<ol style="list-style-type: none"> 1. Textbook materials 2. Lecture 3. Computer (CAI) 	<ol style="list-style-type: none"> 1. Practice Exercises (Textual--off-line) 2. Practice Exercises (on-line) 3. Work through software package labelled cai 	Completed exercises and evidence of completion of program, cai
2. A Systems approach to the Development of Instructional Materials/ Learning Packages	<ol style="list-style-type: none"> 1. Computer (CMI) 2. Learning Packages (off-line) 3. Lecture 	See paper for expansion of this section of figure	
3. Survey of Individualized Instructional Management Systems	<ol style="list-style-type: none"> 1. Films 2. Lecture 3. Computer (CAI) 4. Textual materials 	<ol style="list-style-type: none"> 1. Examination of resources, philosophy, and materials related to each system 2. To compare and contrast two or more systems or critique any one system 	Short paper

SYNTHESIS: Utilizing CW III, the student developed learning package and knowledge of successful instructional management systems; the student, using the test questions in his learning package, develops a mini-CMI system.

FIGURE 2.



FIGURE 3. A student at work in WWSO's terminal laboratory. There are eight IBM 2741 typewriter terminals located in this room with each terminal separated by a semi-partition.

ERIC

increased the student dissatisfaction with the "up"-time on a terminal system which was being shared (sometimes with keen competition) by users from other disciplines. Happily, at this writing, terminals are available to users at WMSC thirty-five hours per week.

Student consciousness and concern resulting from the above two constraints is evident in their comments registered on the Testing Center's evaluation. The following are some actual student comments which have not been edited in any way:

- A. "1. Work load is too demanding for three credit hours. 2. How can you justify, educationally, twenty-four students with eight terminals which are available only two hours daily? 3. More time should be spent on programming the computer. 4. Course has a great deal of potential."
- B. "There should be more access to terminals and credit awarded for this course."
- C. "I showed up several times last week and could not get a terminal. That wasted my time, and I could not finish my assignments. What is the instructor going to do about this?"
- D. "You talk about individualized instruction. Computers should be available when they are needed. Speaking for myself, I had difficulty getting on the terminals because I had other classes during terminal time."

Cited above are just a few comments which resulted from the formal evaluation. Individuals at colleges considering CAI and/or CMI for utilization in course instruction are probably asking themselves, "How could anyone let this happen?" The point is that we did not "let" it happen. Suddenly there we were, afloat without a paddle; but we did manage to survive! The following are some student comments, in addition to those above, which support our claim to survival:

- A. "This course was not only excellent, but it provided me with an opportunity to apply much of the theory learned in other courses to a learning problem. When I began this course, I had both fears and misgivings about the computer. These no longer exist, and I feel that you opened up a new field for me. Do you know of any schools where they have a computer?"
- B. "I would like very much to do some further work in this area. Is there going to be a course we can go beyond this course?"
- C. "I enjoyed the course a great deal. I learned a tremendous amount for a short period of time. I came completely ignorant of computers, and I feel I have left with something. I particularly appreciated the fact that the instructor worked hard and practiced what he preached."

Informal evaluation, which was undertaken after the formal evaluation, was carried out by having a random sample of the students, twelve in all, respond to the following question: "Putting aside the fact that you suffered due to (1) availability of terminals and (2) the difficulty of scheduling your time to the two hours which the terminals were available, how would you rate this course?" These students were given a rating scale of 0=poor; 1=fair; 2=average; 3=good; 4=excellent. The results: Five students rated the course excellent; four students rated the course good; two rated the course average; and one rated the course fair. Most of the subjective comments were similar to those which were offered to substantiate the successfulness of the course.

While data gained from either of the evaluations offers no statistical impact, it did provide the course authors with feedback related to the successfulness of the course which could be utilized in future planning.

Future Plans. Given that both enrollment at the College remains at its present level and that Ed 444 becomes part of an actual program, the following changes and updating will occur: (1) Learning packages will be revised using feedback from students during the summer pilot study. Student comments throughout the course were very strongly in favor of the utilization of learning packages to present course material. Students felt that this use not only provided them with an observable application but also with a model after which to pattern their own packages. (2) The program "system" will be revised to increase reinforcement and provide more information to the student about his score. The rationale for further developing this program is similar to that provided for further development of learning packages to be utilized in the course. (3) A progress/summary table will be provided for the instructors of Ed 444 showing the time spent, score on each criterion-referenced test, etc., for each student. Changes in computer software are now possible because of a new programming language developed at WMSC which permits course authors to use COURSEWRITER III syntax in combination with an interactive problem-solving language (WPL). WPL (Western Programming Language) is a subset of PL/1 which was designed to run in an 88K partition of an IBM 360/Model 40. (4) The actual teaching of CW III will be dropped from Ed 444 in order to lighten the load of the course. Students will take the program "cai" and may, working with the course instructors, independently pursue "hands-on" projects.

Recommendation and Summary. Based on our experience with using the computer in undergraduate curricula in the Department of Education at WWSC, we would like to offer the following recommendations or suggestions to those considering a similar application: (1) Get the support of the Education Department. It is important that the Department make a commitment and recognize that the planning and development of such innovative practices is time-consuming. For example, if one is adding a three-credit hour course of this nature to a normal teaching load, a half-time staff assignment for the course would not be liberal by any means. (2) Establish the course as part of a program leading to a degree or certification. Do not develop it as an elective unless student and faculty interest is such that an adequate number of students can be solicited to offer the course at least twice a year. To put this amount of work into a course and have it shelved would be a shame. (3) Assure the availability of terminals so that students can exercise freedom to use the computer when they have time. This is one of the basic advantages to CAI learning, and particularly the first encounter of a student with the computer should be an enjoyable one. (4) With the projected growth of computers in this country, 125,000 additional computers by 1980[2], utilization of the computer in undergraduate education is a must. The utilization of the computer in classroom instruction in K-12 education in this country has increased such that departments of education at institutions of higher learning are shirking their responsibilities if they do not expose future teachers to the computer as a medium for instructional communication[3]. (5) Rather than consider a course which involves students' learning and applying a programming language, consider utilizing the computer as a medium within the course presentation. The latter introduces the student to the computer in the ultimate, for he is provided the opportunity to see an instructor's actually making use of the computer. It has been said that "one does as others do" and that teachers teach using the methods by which they were taught. If one is to accept this premise, then the above application may be more important than actual courses which teach CAI.

In summary, we would like to suggest that education departments in colleges throughout this country consider carefully the use outlined in Recommendation 5. It is here that we suggest utilization of the computer in undergraduate curricula in the department of education begin. Gaining experience and success from this initial application will make the next step, courses in CAI and CMI, an easy transition. Education 444 was successful and has proved worth pursuing. The authors of this paper view the frank appraisal of experiences discussed throughout this document as a means of moving forward.

REFERENCES

1. Latta, Raymond F., and Papay, James, "Planning for Change: An Iterative Approach", Planning and Changing: A Journal for School Administrators, Number 2, July, 1971, pp. 70-80.
2. "Will We Call Them the Sensitive Seventies?", Administrative Management. January, 197, p. 22.
3. Dick, Walter; Latta, Raymond F.; and Rivers, Leroy, "Public School Activities and Related Sources of Information Concerning CAI in United States", Educational Technology. Number 3, March, 1970, pp. 36-39.

A COMPARISON OF TYPES OF FEEDBACK TO
STUDENT RESPONSES IN A CAI UNIT

Harold L. Schoen
Virginia Polytechnic Institute
and State University
Blacksburg, Virginia 24061
Telephone: (703) 951-2471

Introduction

There is much resistance in education to the use of tutorial computer assisted instruction in the classroom. The main objections seem to be:

- a. it costs too much, and
- b. it is not an improvement over existing modes of instruction.

At present, both objections appear to be valid. However, the first is probably a temporary condition. As schools purchase more computer time for management and problem solving uses, the on-line cost may well become an insignificant factor. At any rate, this paper is directed to the second objection.

The question of how individualization of instruction via CAI can best be achieved is "an almost untouched problem" (Gentile, 1967). This study is an attempt at "touching" that problem. Four instructional treatments which differed in the degree of individualization and personalization (references to the student by name) in a CAI program designed to teach the mathematical concept of function were written. Mean achievement and attitude scores of groups of students receiving the different treatments were then compared.

Background

Little research or development in CAI had occurred prior to 1964 (Dick, 1965; Suppes, 1966). One study conducted at I.B.M., which compared achievement scores of students learning a topic via three modes of instruction, CAI, a standard classroom approach, and programmed instruction, showed results which strongly favored the CAI treatment. However, the Hawthorne effect of novelty was probably great (Dick, 1965).

In another early study two groups of high school students received instruction in logic from a computer. One group received a fixed sequence of items while subjects in the other were branched depending on their performance during the lesson. The posttest achievement scores were significantly higher (.05 level) for the branching group than for the fixed sequence group (Coulson, 1962).

Some trends which early research and development in CAI seemed to indicate were that:

- a. the type of error a student make must be indicated to him,
- b. the CAI program must provide help to correct student errors, and
- c. this help should be given immediately following the error (Dick, 1965).

Problems which became apparent in early CAI development were:

- a. development and evaluation of better teaching strategies to utilize the capabilities were needed, and
- b. programs designed to record student responses and present them for analysis in rapid and intelligent fashion were not available (Filep, 1967).

The second of these problems has been virtually solved as was exemplified in the data collection for this study.

CAI research and development in the last four or five years has been much more extensive than previously. The recent research studies which are cited have been restricted to those attempting to improve CAI teaching strategies since that is the main thrust of this investigation.

It was found in a study utilizing a four hour CAI course in problem solving that achievement scores were affected by certain personality traits coupled with the type of CAI treatment. One group of subjects took the computer course individually at the terminal while subjects in another group took it with a partner. The subjects who tended to have dominant personalities achieved more (as measured by the posttest) if they worked alone. Those who tended to fear tests achieved more with a partner. Several other personality traits were considered with no significant differences in test scores (Sutter, 1967). Further findings tend to support these results (O'Neill, 1969; Majer, 1970; Gallager, 1970).

No significant differences in learning or retention were found among groups receiving five different types of feedback on a CAI unit which taught science concepts to university upperclassmen (Gilman, 1967; Gilman, 1969). Significant differences were not found in achievement scores in another study comparing three modes of instruction - two modes of CAI and one conventional classroom mode (Proctor, 1969). The relative effect of verbal definitions and numerical examples as corrective feedback in a computer assisted learning task was investigated by Keats and Hansen (1970). Results favored the use of verbal definitions.

Several studies were located which tend to significantly favor the use of CAI for various instructional tasks. These include CAI over programmed instruction (Dick and Latta, 1970), the value of CAI for testing students (Ferguson, 1969), CAI over the traditional classroom instruction (Strojkiwicz, 1968), and the value of CAI to teach geometric topics (Dennis, 1968). In another study, however, two modes of CAI were compared - one consisting of multiple choice questions and the other of a series of lecture-like presentations - with no significant achievement differences resulting (Bauldree, 1968).

Grayson (1970) summarizes the state of the art in CAI research,

While many (CAI) studies have been conducted ... in many of them, the Hawthorne effect of novelty may be the overwhelming factor.

This problem notwithstanding the research does seem to show that CAI has potential as a mode of instruction, and that the individual student must be studied in an attempt to meet his needs. However, no clearly superior approaches to CAI development appear to be emerging.

Method

Two CAI units were developed by the researcher - Unit A, including concepts prerequisite to functions such as sets, ordered pairs, and graphs; and Unit B, containing the definition of function, graphs of functions, and functional notation. Unit A and Unit B were written in I.B.M. Coursewriter III, version 2, and are available on one of The Ohio State University's I.B.M. 360/50 computer systems. Four types of feedback to incorrect student responses were written in Unit B. These are the result of crossing two levels of each of two variables, I (individualization) and P (personalization).

The levels of I were defined as follows.

- I' - the student receives feedback following an incorrect response which states why his answer is incorrect and gives the correct response.
- I'' - the student receives feedback following an incorrect response which states that the given answer is incorrect and gives the correct answer with a reason why it is correct, but the feedback does not refer specifically to the student's response.

The feedback to correct student responses was the same for the two levels of I. The lengths of the feedback statements for the two levels were as nearly equal as possible.

The levels of P were defined as follows.

- P' - the student's first name appears in some of the feedback to both correct and incorrect responses. The frequency of use of the first name was decided by what seemed reasonable to the researcher. The only pattern followed for use of the first name was that if the name was used in feedback to one

response to a question it appeared in feedback to any response to that question. Hence, the student's error rate did not affect the frequency with which his first name appeared. To be exact, the name appeared in feedback to responses to 41 of 56 questions in Unit B.

Pⁿ - the student's name never appeared in the feedback.

The four types of feedback then were I'P', I'Pⁿ, IⁿP', and IⁿPⁿ which result from crossing the two levels of I with the two levels of P.

As an example of the type of feedback for each group, suppose this question is asked the subject.

So in the previous question, $S = \{(x,y): y = 2x + 3 \text{ and } x = 1, 2, \text{ or } 3\}$ is a function.

- whose domain is [9, 7, 5] and whose range is [1, 2, 3]
- whose domain is [1, 2, 3] and whose range is [9, 7, 5]
- whose domain is x and whose range is $y = 2x + 3$
- none of the above

If the student response is c then the I'P' feedback is:

x is only the symbol representing the values in the domain and y, those in the range. The correct answer is b, Edgar.

The I'Pⁿ feedback is the same except 'Edgar' is omitted.

The IⁿP' feedback is:

No, the domain is the set of values x (or the first coordinate) may take, so the domain is [1, 2, 3]. The range is the set of y values so $y = 2x + 3$ has values $2(1) + 3 = 5$, $2(2) + 3 = 7$, and $2(3) + 3 = 9$. The correct answer is b, Edgar.

The IⁿPⁿ feedback is identical except 'Edgar' is omitted. Feedback for the two Iⁿ groups is the same for any incorrect response; that is, c, a, or d.

The CAI units were based on a set of researcher developed behavioral objectives which followed the recommendations of several authors (Bohein, 1968; Gagne, 1964; Gagne and others, 1965; Krathwohl, 1964; Lindvall, 1964; Hager, 1962; Tyler, 1964; Ullmer, 1967). The branched programmed CAI units were designed to lead the students to achieve the behavioral objectives. Three achievement tests, Q, R, and S, and an attitude scale, T, were developed (twenty items of the thirty-item attitude scale were used with permission of Carlton Robarney, Michigan State University). Two pilot studies for purposes of revising and refining the instructional units and evaluation instruments preceded the final study. Q was a twenty-item test of Unit A, R was a fifteen-item test of Unit B, subunit 1 (the definition of function), and S was a fifteen-item test of Unit B, subunit 2 (functional notation). Q, R, and S were composed of multiple choice items. The number of items and the material tested by each instrument were decided based upon pilot data.

Sixty pre-calculus mathematics students at Ohio State University in the Winter Quarter, 1971, were the subjects. They comprised two sections of Math 150 (students are placed into sections at Ohio State from a shuffled deck of class cards). The subjects were then placed randomly in equal numbers into the four treatment cells. During the first three weeks of the quarter all subjects were administered Unit A followed immediately by Test Q, then Unit B, subunit 1, followed immediately by Test R, and finally Unit B, subunit 2, followed immediately by Test S, all via an I.B.M. 2741 terminal. Two days after all subjects had received the computer treatment, they completed T, the attitude scale, using pencil and paper. The Unit A treatment was included prior to the experimental unit on which the comparisons were made (Unit B) in order to decrease the Hawthorne Effect of novelty. The Unit A treatment was identical for all four treatment cells.

The null hypotheses of no differences between mean scores of the combined group I'P' and I'Pⁿ compared with the combined group IⁿP' and IⁿPⁿ and of no differences between mean scores of the combined group I'P' and IⁿP' compared with the combined group I'Pⁿ and IⁿPⁿ on each of the criterion measures R, S, and T were posed. Nearly equal cell means prompted eliminating the use of Test Q scores as a covariate. Thus the hypotheses were tested using two-way analyses of variance with R, S, and T scores as the respective dependent variables. The BMD05V program for testing general linear hypotheses was used.

Results

The results of the main comparisons are summarized in the following table.

Using R scores as the dependent variable the I effect is clearly non-significant, but the P effect is significant with p less than .11. With the small N in each cell, this may be an indication of some differences in the levels of P; namely, that the P' groups, those receiving feedback with student names included, tended to perform better than the P" group on R.

The results of the analysis of variance using S scores as the dependent variable indicate that students receiving the I" treatment scored higher (p less than .04). This is in contrast to nearly equal means on Test R for the two groups. Analyzing the P effect shows that the P' group, those receiving their names in feedback, scored higher than the P" group, though not significantly higher. The reliability of the fifteen-item Test R was estimated by $KR - 20 = .70$ ($N = 60$).

To analyze the T (attitude) scores, student responses were coded from 0 to 4 with a higher score indicating a more positive attitude. The results in the Table are based on student scores computed by taking the sum of the coded student response for each of the thirty items. Using correct to mean a 3 or 4 on an item and incorrect to mean a 0, 1, or 2, the reliability of this thirty-item instrument was estimated by $KR - 20 = .84$, where $N = 58$. Here N is 58 instead of 60 because two subjects dropped the course before the administration of T. The attitude scores of students receiving the P' treatment are higher than those receiving the P" treatment at the .07 probability level, while the I effect is clearly non-significant.

Further analysis showed non-significant correlations between attitude and achievement and small positive correlations between Unit B time and attitude scores. Mean student CAI treatment time was 171.07 minutes. This includes the time taken to complete Unit A and Unit B as well as tests Q, R, and S. Student time did not differ significantly among the treatment cells.

Conclusions

The results of this study do not suggest that the type of individualized feedback (I') written for this program yields better student achievement or attitudes scores than non-individualized feedback (I"). In fact, on one achievement test, scores were significantly higher in the I' group, while no significant differences in the levels of I were found in the other achievement test scores or the attitude scores. These results, at first glance, seem to imply that a statement of the correct answer with a reason why it is right is better corrective feedback than an individualized explanation.

A closer look at the individualization levels, however, suggests that subjects in the I' group were not ready to continue to the next frame following their corrective feedback while those in the I" group were. An incorrect answer to a question, certainly in most cases, implied that the subject did not understand the concept treated in that question. The I' feedback explained the fallacy in the incorrect answer and stated the correct answer. Then the student went to the next frame even though no reason for the correct answer was given. On the other hand, the erroneous student who received the I" feedback was given an explanation concerning the correct response. The explanation of the correct answer was probably the key difference in the I levels of feedback.

The use of a student's first name in the CAI program seemed not only pleasing to him (attitude difference at p less than .07), but there was also some evidence that better achievement occurred when the first name was used. The positive correlation between Unit B time and attitude scores seems to indicate that the students who enjoyed the program most did not hurry to finish, while those who disliked it rushed through. This result may have been caused by a few students who became frustrated because they did not understand the explanations or were simply tired.

In summary, the results of this study seem to imply that:

- a. an important component of corrective feedback is an explanation of the concept in question if that explanation immediately precedes a new frame,
- b. the attitude of students toward a CAI tutorial program is improved by use of their names in the program, and
- c. there is a direct relationship between student on-line time and attitude toward a CAI program (provided that the program is designed so that a student error rate does not greatly affect his progress through the unit).

Any conclusion concerning the desirability of individualized corrective feedback does not seem warranted. More research is needed to answer this important question.

Future Research

Some research related to the question of how to individualize CAI tutorial programs is presently being planned. A graduate student at The Ohio State University, Kenneth Taylor, has proposed a study which will replicate the design of this one, with a different CAI program and a different sample population. His program will be designed to teach geometric concepts to college seniors.

A follow-up of this study in which more feedback treatments are considered is now being designed by the investigator. A combination of the I¹ and I² feedback is planned, adding a third level to the individualization variable. It is hoped that this study can be done with a sample of high school juniors or seniors since the results could then be generalized to a larger population than those of the reported study.

REFERENCES

- Bauldree, A. M. Individual and group differences in learning under two different modes of computer-assisted instruction. (Doctoral Dissertation, Florida State University) Dissertation Abstracts, 1968, 29, 1443-A. (Abstract)
- Bohein, J. Implications of the individualization of instruction for curriculum and instructional design. Audiovisual Instruction, 1968, 68, 13, 238-242.
- Coulson, J. E., Estavan, D. P., Melargno, R. J., and Silberman, H. P. Effects of branching on a computer controlled autoinstructional device. Journal of Applied Psychology, 1962, 46, 389-392.
- Dennis, J. R. Teaching selected geometry topics via a computer system. (Doctoral Dissertation, University of Illinois) Dissertation Abstracts, 1968, 29, 2145-A. (Abstract)
- Dick, W., and Lalta, R. Comparative effects of ability and presentation mode in computer-assisted instruction and programmed instruction. AV Communication Review, 1970, 18, 33-37.
- Dick, W. The development and current status of computer-based instruction. American Educational Research Journal, 1965, 2, 41-54.
- Ferguson, E. L. The development, implementation, and evaluation of a computer-assisted branched test for a program of individually prescribed instruction. (Doctoral Dissertation, University of Pittsburgh) Dissertation Abstracts, 1970, 3856-A. (Abstract)
- Filep, E. T. Individualized instruction and the computer potential for mass education. AV Communication Review, 1967, 15(1), 102-113.
- Gagne, R. M. The implications of instructional objectives for learning. In C. E. Lindvall (Ed.) Defining Educational Objectives. Pittsburgh, Pa.: University of Pittsburgh Press, 1964.
- Gagne, R. M. and others, Psychological principles in system development. New York: Holt, Rinehart and Winston, Inc., 1965.
- Gallagher, P. D. An investigation of the relationship of learning characteristics and success in a computer managed instruction course for graduate students. Paper presented at the meeting of the American Educational Research Association, April, 1970.
- Gentile, J. R. The first generation of CAI systems--an evaluative review. AV Communication Review, 1967, 15(1), 23-54.
- Gilman, D. A. Feedback, prompting, and overt correction procedures in non-branching computer assisted instruction programs. Journal of Educational Research, 1967, 60, 423-429.
- Gilman, D. A. Comparisons of several feedback methods for correcting errors by computer-assisted instruction. Journal of Educational Psychology, 1969, 60, 503-508.
- Grayson, L. P. A paradox: the promises and pitfalls of CAI. EDUCON, March, 1970, 1-3.

- Keats, J. G., and Hansen, D. M. Definitions and examples as feedback in a CAI stimulus-centered mathematics program. Unpublished manuscript, Florida State University, 1970.
- Krathwohl, D. R. The taxonomy of educational objectives--its use in curriculum building. In C. M. Lindvall (Ed.), Defining Educational Objectives. Pittsburgh, Pa.: University of Pittsburgh Press, 1964.
- Lindvall, C. M. The importance of specific objectives in curriculum development. In C. M. Lindvall (Ed.), Defining Educational Objectives. Pittsburgh, Pa.: University of Pittsburgh Press, 1964.
- Mager, R. F. Preparing Instructional Objectives. Palo Alto, California: Fearon Publishers, 1962.
- Majer, K. S. A study of computer-assisted multimedia instruction augmented by recitation sessions. (Doctoral Dissertation, Florida State University) Dissertation Abstracts, 1970, 30, 3641-A. (Abstract)
- O'Neill, H. F. Effects of state anxiety and tasks difficulty on computer-assisted learning. Journal of Educational Psychology, 1969, 60, 343-350.
- Proctor, W. L. A comparison of two instructional strategies based on computer-assisted instruction with a lecture-discussion strategy for presentation of general curriculum concepts. (Doctoral Dissertation, Florida State University) Dissertation Abstracts, 1969, 29, 2075-A. (Abstract)
- Strojkeiwicz, L. W. Training for problem-solving skills utilizing a computer-assisted instructional method. (Doctoral Dissertation, University of California at Berkeley) Dissertation Abstracts, 1968, 29, 1498-B. (Abstract)
- Suppes, P. The uses of computers in education. Scientific American, 1966, 215, 206-220.
- Sutter, E. M. Individual differences and social conditions as they affect learning by computer-assisted instruction. (Doctoral Dissertation, University of Texas) Dissertation Abstracts, 1967, 28, 4012-A. (Abstract)
- Tyler, E. W. Some persistent questions on the defining of objectives. In C. M. Lindvall (Ed.), Defining Educational Objectives. Pittsburgh, Pa.: University of Pittsburgh Press, 1964.
- Ullmer, E. J. A study in the development of a technology based model for instruction design. (Doctoral Dissertation, University of Wisconsin) Dissertation Abstracts, 1967, 28, 4551-A.

USING A COMPUTER TO SUPPORT THE TESTING PROGRAM IN AUDIO-TUTORIAL BIOLOGY

Arthur Wagner and Ronald Bleed
Joliet Junior College
Joliet, Illinois 60436
Telephone: (815) 729-9020

Introduction

In courses of study employing individualized instruction such as the audio-tutorial biology program at Joliet Junior College, one is faced with the problem of producing multiple versions of objective tests for each minicourse in order to maintain the integrity of the testing program. It is a particular problem because the objective test is necessarily short (10 questions, multiple choice) in order to provide time during the testing session for the oral quiz. Students easily learn the answers for this short test from others who have already taken it. Subsequent evaluation of this student's progress then becomes quite invalid. The obvious solution was to make a different version for each of the 40 testing sessions given for each minicourse. All versions of the test had to be as similar as possible with regard to difficulty and representation of the subject matter. Forty test versions had to be produced for each of the 15 minicourses presented in the semester.

It was soon apparent that the production of the tests was beyond the capacity of the biology department's secretarial resources, and the biology staff sought out the data processing department and the computer for assistance. Together, they developed an operation which not only solved the problems at hand but also increased the capabilities for testing beyond earlier expectations.

Developing the Test

The biology staff wrote 4 questions at the knowledge level of Bloom's taxonomy (Bloom, 1956) for each behavioral objective in the minicourse. These formed a question pool from which were drawn individual questions for each version of the test. Two criteria were established for the selection of questions for each test.

1. Questions for each test were selected on a random basis.
2. The random selection was controlled to insure the equal distribution of questions among the objectives of the unit.

Producing the Test

After the questions were compiled for each test, they were produced in multiples of 8 to accommodate our testing methods (Postelthwait, 1968). An answer sheet was also produced for each version of the test in the same format as the question sheet. To the left of each question on the answer sheet was given the minicourse number, the behavioral objective number which each question was based, and the correct answer. Figure 1 shows one question from such an answer sheet.

The answer sheet expedites feedback to the student from the instructor immediately after the test is completed and graded not only about the question missed but, more importantly, to the specific behavioral objective upon which it was based. This provides him with explicit information to guide preparation for a retest.

Using the Computer

Except for writing the initial test questions for each behavioral objective in every minicourse, all of the other operations were computerized, including the printing of the multiple copies of each test and its accompanying answer sheet. The initial step was to keypunch the test questions and possible responses. A card format was designed to permit any type of question, objective, short answer or essay, any length of question and any number of responses. Columns 1-3 were used for question number, column 4 for line number, column 5 for minicourse number, column 6 for objective number, column 7 for answer, and columns 8-70 for the text of the question or response.

Substituting, adding, or deleting questions can easily be accommodated by this system. This format also made it possible to re-sort the cards if they became disordered. The cards were stored in groups by minicourse which permitted the easy selection of any minicourse for test preparation. For proficiencies or final exams, the entire deck was selected.

- 3 Darwins experiences on the HMS beagle led him to conclude that —
- ③②* A. Organisms inherently demonstrate variations.
 B. Organisms must change when the need arises.
 C. Organisms, only which are the fittest, can compete.
 D. Organisms, respond mostly to genetic drift rather than mutations.

○ = minicourse number
 □ = behavioral objective number
 * = correct answer

FIGURE 1. A sample question from a typical answer sheet keyed to a minicourse and behavioral objective.

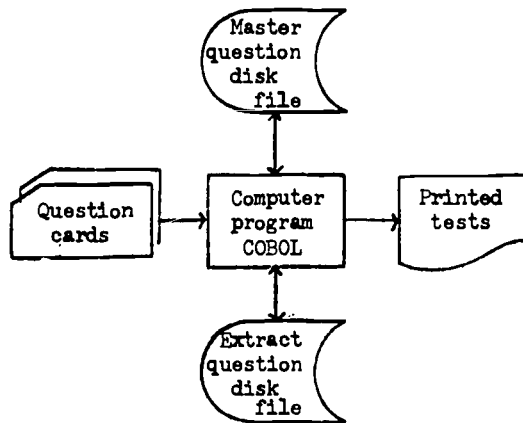


FIGURE 2. The computer system flow chart.

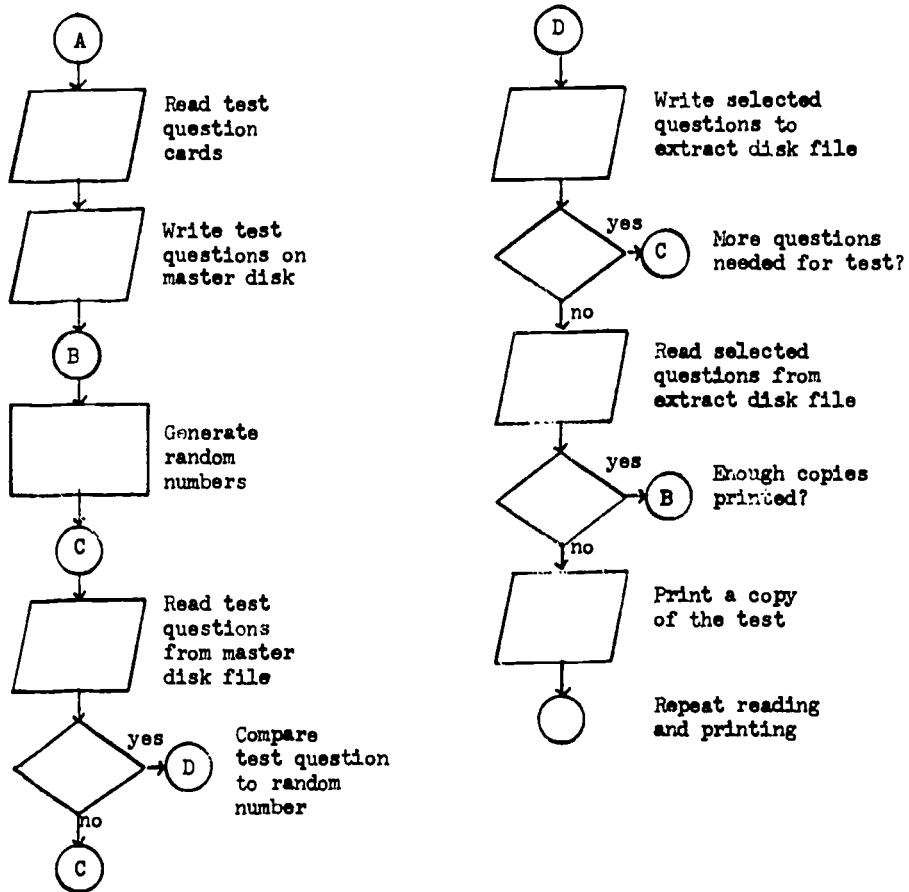


FIGURE 3. The computer program flow chart.

The computer system involves only 1 program written in standard COBOL and run on a small NCR Century 100 computer. The system flow chart is shown in Figure 2.

The first step in the computer program reads the cards for the selected minicourse and writes the card records on a disk in the form of a master file. Second, a series of random numbers is generated that equals the number of questions required and evenly distributes the question among the objectives. Third, the master file is searched for the questions matching the random numbers. When a match is found, the question from the master file is written to an extract disk file. This step is repeated until all selected questions have been written to the second file. Fourth, the second disk file is used as input to print out the answer sheet and finally to print out the required number of question sheets. Upon the completion of the test set for this individual version, the program branches back to the second step and repeats the same procedures for another version. This is illustrated in Figure 3.

This system is advantageous for courses which require multiple versions of many tests on a regular basis. The audio-tutorial and programmed instruction methods lend themselves to this type of test preparation. It would be quite difficult and expensive to duplicate this without a computerized system.

SUMMARY

To facilitate the testing program for an audio-tutorial system in the biology department, a method was devised to generate different tests for each of the 40 small test sessions. Each test fulfilled the following requirements.

1. Each had different, randomly selected questions.
2. Each had an equal distribution of questions among the objectives for the unit.
3. Each had the capability of having any number of questions.
4. Each had an individual answer sheet with the correct answer marked and the minicourse and objectives labeled.
5. Each had the capability of being printed in any number of student copies.
6. Each had to be capable of handling any type of question of any length.

To fulfill these requirements, the biology department looked to a computer. Working with the data processing center, a small system was designed, and a computer program was written in COBOL which could generate the tests according to the requirements.

REFERENCES

1. Bloom, B. S., Ed., Taxonomy of Educational Objectives, New York: David McKay Company, 1956.
2. Postelthwait, S. N., Novak, J., Murray, Jr., H. T., The Audio-Tutorial Approach to Learning, Minneapolis: Burgess Publishing Co., 1969.

INITIAL DEVELOPMENT OF INDIVIDUALIZED
INSTRUCTION WITH COMPUTER SUPPORT

Richard P. Walters, Gail M. Nishimoto
John M. Horowitz, Ray E. Burger
University of California
Davis, California
Telephone: (916) 752-3241

Effective use of computers in the learning process can be achieved only after a great deal of patient preparation, testing, evaluation and evolution. The road already travelled blurs rapidly for those whose sights are set on the new horizons opened by achievements already secured. These visionaries will naturally wish to share their most recent successes and to chart the road ahead; their guidance is essential. For those who follow in turn, painfully tracing the fading steps of the pioneers in this field, it often seems that the gap is too great, the signposts too few, and the pitfalls too numerous ever to attain those plateaus considered old hat by the experts. This paper is an attempt to add a few signposts to the beginning of the road, in the hopes that effective computer support of the learning process may become a major highway rather than a limited thoroughfare. More specifically, this report describes a series of simple programs that provide self-evaluation for the student, course review for the instructor, and an opportunity to proceed into more sophisticated computer-supported instruction.

Instructional use of computers has developed slowly for several reasons, including problems of developing adequate hardware and software support, dollar costs of entry and continuation in computer support, incomplete understanding of the most appropriate methodologies suitable for computer technology, the time required for significant educational returns on the initial preparation of instructional material and a significant lag in the acceptance of computer approaches to learning. Although all these factors play a role in computer use for instruction, the problems of operational cost, methodology and short-term returns offer the greatest promise for creative solutions.

Operational costs of a time-shared system can be allocated to the computational steps performed, the storage of information in machine-readable form, and data communication. The additional costs of timesharing can be divided between the computation and communication costs of the system. The major opportunities for realizing savings exist in the use of teaching strategies that minimize storage of large files. Additionally, the time spent by a student at a remote terminal dictates the number of terminals, communications lines and entry ports that the system must provide for a class. Some cost savings can therefore be realized by restricting the student's computer time to those experiences offering learning benefit.

The instructor who seeks to use a computer as a teaching aid is confronted with a need not only to familiarize himself with the unique characteristics of the computer, but also to define with considerable precision his teaching objectives so as to utilize this auxiliary technology effectively. It often happens that the computerization of a learning segment is undertaken without adequate planning, and the results are usually unfortunate. Furthermore, when the instructor has his objectives well in mind he may not be sufficiently aware of the computer technology to take full advantage of the versatilities available to him.

A major factor affecting the initiation of computer-related instruction is the time required for a system to deliver effective learning experiences. A typical first encounter with computer implementation of a learning sequence would begin with systems problems when the equipment is first installed, followed by a protracted period of program creation and implementation, then followed by a still longer time for preliminary testing of the material, evaluation and modification prior to introduction in a course. Attempts to by-pass this sequence by acquiring available programmed instruction usually fail to involve the instructor sufficiently in the development process to educate him to the merits and weaknesses of this technique.

As a result of the factors described above, persons wishing to introduce computer technology in the instructional process find themselves on the horns of a dilemma, faced with tradeoffs between effective instruction, inexpensive implementation and the time required for startup. There is no single solution to this problem; however, the technique described below appears to serve as an effective entry point without major drawbacks in any of the problem areas cited above.

Historically, the Davis campus of the University of California had done little to implement on-line computer support prior to 1970. The major step that had been taken was to acquire a third generation computer system capable of supporting timesharing, and to back that system up with an existing timesharing system during conversion to the new equipment. In addition, some

attempts to explore instructional use of computers had been initiated, notably within the school of medicine and the department of electrical engineering. Animal physiology, on the other hand, had been exploring the merits of individualized instruction from other methodological approaches, aimed at increasing the opportunities for independent study and self-evaluation. In the summer of 1971, the department of animal physiology and the Office of Medical Education decided to attempt a joint development program that would enhance self-evaluation for both departments.

The faculty of the department of animal physiology were faced with rapidly increasing enrollment in an already large undergraduate class in systemic physiology. In order to maintain instructional quality, these faculty had evolved a team teaching concept of optional lectures supplemented by audiovisual material containing the same concepts, by self-tests administered with a hand-scored see-through board and by small tutorial sessions. The School of Medicine also offers team-taught courses, interdisciplinary in nature, as a part of its regular core curriculum. The course committees for these courses include representatives of the Office of Medical Education who have responsibility for the delivery and evaluation of courses and also for the development of new teaching methodologies, including computer support. Here, too, students were anxious to receive increased self-evaluative experiences. The opportunity to combine forces appeared favorable, and as a result the two schools decided to work together in designing a single program for self-evaluation. Design of the methodology was accomplished in joint sessions; the program was introduced simultaneously in both schools, using slightly different formats as described below.

Description of Self-Evaluation System

The computer supported self-evaluation system provides for the entry of answers to problem sets so that students can review these tests by means of a computer terminal, receiving immediate tutorial guidance while their responses are stored in a file available for review by the instructor. Three interactive computer programs were written to support this process: the first for entering the answer key, the second to be used by the student in taking the self-test, and the third for use by the instructor to evaluate overall student performance in a given course segment. In addition, the programs use two disk files, one containing student identification and the second a set of comments directed to individual students.

In entering the answer key (fig. 1) the instructor (or a teaching assistant) is first asked to provide a title for the self-test, and then to specify certain procedural options such as the type of identification required and the decision to provide or withhold the correct answer. The next step is to enter correct answers to each question. The program was designed to eliminate computer storage of questions for several reasons. Since questions can be easily typed and duplicated for distribution, keypunching and storage of the questions are unnecessary and expensive. The time required for a student to consider individual problems is not productive in terms of the man-machine interaction, and it might be better accomplished away from the terminal so as to save on connect costs as well as freeing the terminal for other students' use. In contrast, however, the answer options are more flexible than the normal multiple choice tests typical of most computer-scored experiences: they may be either numeric or include combinations of alphanumeric characters. Numeric answers are considered correct if they fall within a specified range of the "correct" response. Typically, an instructor will use a mixture of numeric, true/false, single word and multiple choice responses in a given test.

Following entry of the answers, the instructor then enters feedback responses for incorrect answers. These responses may direct the student to specific reference material, contain suggestions for review or instruct the student to contact a tutor for clarification. The responses may be recalled on the basis of a single incorrect answer or a combination of errors, depending on the instructor's judgement.

At the conclusion of the instructor's session, there will be stored on disk a series of answers, tutorial comments for the combinations of incorrect answers, and a separate file of personal comments for students. At this point the problem set is ready for the second step, during which the student enters his answers. Prior to his arrival at the terminal, each student will have received a problem set which he is instructed to review on his own before talking it over with the computer. He is usually encouraged to use those auxiliary aids appropriate for the course segment and to delay his session at the terminal until he is reasonably satisfied that he understands each question. When he signs on, giving his name and/or his code number, any comments stored under his name will be displayed before he enters his answers. The program then requests the answers to the problem set, giving the student an opportunity to review and correct typographical or other errors before they are scored (fig. 2). When the student indicates that he is satisfied with his answers, his performance is evaluated. He is given his overall score, informed as to which questions he missed, and then provided the tutorial information appropriate to his incorrect responses. He is then free to leave the terminal. Usually a student will require five minutes or less for a set of twenty to thirty questions.

RUN EXAM

-80J- OEXAM
YOU ARE ABOUT TO ENTER THE ANSWERS FOR A QUIZ.
PLEASE ANSWER ALL QUESTIONS THAT I ASK, SO THAT
I CAN QUIZ THE STUDENTS ACCORDING TO YOUR WISHES.
EACH TIME I EXPECT A RESPONSE FROM YOU I WILL TYPE
A "?". PLEASE WAIT FOR IT TO APPEAR BEFORE YOU TYPE
YOUR RESPONSE. PLEASE END EACH RESPONSE WITH A
SHIFT/EXIT. IF YOU MAKE A MISTAKE AND HAVE NOT HIT
THE SHIFT/EXIT THEN USE THE LEFT ARROW TO BACK UP
TO THE WORD IN ERROR AND CORRECT BY USING THE INSERT
CHARACTER KEY (I/C), DELETE CHARACTER KEY (D/C),
OR RETYPE FROM THAT POINT ON.

ENTER EXAM TITLE OR TYPE "STOP"
?PHYSIOLOGY REVIEW #1

WHAT IS THE MAXIMUM NUMBER OF TIMES A STUDENT
MAY TAKE THE EXAM?
?1

NUMBER OF QUESTIONS?
?4

ANSWERS, PLEASE

- 1. ?VENTILATION
- 2. ?TRUE
- 3. ?A
- 4. ?2.43

IF YOU WISH TO MAKE ANY CHANGES IN THE ANSWERS
TYPE QUESTION NUMBER, SPACE AND NEW ANSWER.
YOU MAY MAKE AS MANY CHANGES AS YOU WISH, WHEN DONE
TYPE STOP.

WOULD YOU LIKE ME TO GIVE THE STUDENT THE
CORRECT ANSWERS FOR THOSE QUESTIONS HE
ANSWERS INCORRECTLY? (YES OR NO)
?YES

WILL STUDENT GIVE NAME, NUMBER OR BOTH?
(TYPE "NAME", "NUMBER", OR "BOTH".)
?BOTH
WILL NAMES AND/OR NUMBERS BE SUPPLIED FOR THOSE STUDENTS WHO
WILL PARTICIPATE?
?YES

WOULD YOU LIKE COMMENTS DISPLAYED IF
STUDENTS MISS QUESTIONS? (YES OR NO)
?YES

FOR EACH COMMENT YOU ENTER, YOU MUST GIVE AN
"OR" STRING AND AN "AND" STRING.

A STRING IS A LIST OF QUESTION NUMBERS
SEPARATED BY SPACES. (E.G. 1 2 5).

IF IT IS AN "OR" STRING THEN THE COMMENT
WILL BE DISPLAYED IF THE STUDENT MISSES ANY ONE
OF THE QUESTIONS LISTED.

IF IT IS AN "AND" STRING THEN THE STUDENT
MUST MISS ALL QUESTIONS IN THE LIST FOR THE
COMMENT TO BE DISPLAYED.

YOU MAY ENTER AS MANY COMMENTS FOR DISPLAY
AS YOU WISH, BUT EACH MUST HAVE BOTH TYPES OF STRINGS.
(IF YOU ONLY WISH TO GIVE ONE STRING, HIT SHIFT/EXIT
WHEN THE OTHER IS REQUESTED.)

ENTER "OR" STRING FOR COMMENT 1

? 1 4

ENTER "AND" STRING FOR COMMENT 1

? 1

ENTER TEXT OF COMMENT 1 END TEXT WITH AN ASTERISK (*).

?0.1 OR 0.4 A REVIEW OF BASIC RESPIRATORY PROCESSES IS ADVISED.

? PLEASE SEE CHAPTER 5 OF TEXT.*

MORE COMMENTS?(YES OR NO)

?YES

ENTER "OR" STRING FOR COMMENT 2

? 1

ENTER "AND" STRING FOR COMMENT 2

? 1 2 3

ENTER TEXT OF COMMENT 2 END TEXT WITH AN ASTERISK (*).

?0.2 AND 0.3 PLEASE REVIEW CHAPTER 3 ON ACTION POTENTIAL.*

MORE COMMENTS?(YES OR NO)

?NO

FIGURE 1

RUN GETIT

-B0J- 0GETIT

WHAT IS YOUR NUMBER AND NAME?
(NUMBER, SPACE, LAST NAME, SPACE, FIRST NAME)
?30 CASEY KAREN

KAREN, SEE ME IF YOU NEED HELP BEFORE THE NEXT EXAM. DR. H
PHYSIOLOGY REVIEW #1

INTRODUCTION NECESSARY? (YES OR NO)
?N0

PLEASE ANSWER ALL 4 QUESTIONS.

1. ?VENTILATION
2. ?FALSE
3. ?B
4. ?2.43

WHICH OPTION?
D FOR DISPLAY.
C FOR CHANGES.
N FOR NO CHANGES AND NO DISPLAY.
HELP FOR HELP.

?N

QUESTION	GIVEN ANSWER	2 QUESTION(S) INCORRECTLY.	CORRECT ANSWER
2	FALSE		TRUE
3	B		A

SEE THE FOLLOWING REFERENCES AND COMMENTS FOR HELP. I WILL PAUSE
AND TYPE A ? BEFORE EACH COMMENT. WHEN YOU ARE
READY FOR ME TO PROCEED, HIT SHIFT/XMIT.

Q.2 AND Q.3 PLEASE REVIEW CHAPTER 3 ON ACTION POTENTIAL.

SORRY, KARENBETTER LUCK NEXT TIME.

YOUR SESSION HAS BEEN COMPLETED.
THANK YOU

FIGURE 2

R TELLME

-BOJ- OTELLME

WOULD YOU LIKE RESULTS PRINTED (P) AT COMPUTER CENTER OR
DISPLAYED (D) AT YOUR TERMINAL NOW? (P OR D)
?D

QUIZ WAS TAKEN 3 TIME(S).

PLEASE TYPE ONE OF THE FOLLOWING:

- 1 FOR INDIVIDUAL STUDENT RESULTS.
- 2 FOR FREQUENCY DISTRIBUTION OF GIVEN ANSWERS.
- 3 FOR TOTAL NUMBER OF INCORRECT RESPONSES FOR EACH QUESTION.
- 4 FOR BOTH 1 AND 2.
- 5 FOR BOTH 1 AND 3.

?4

STUDENT HAS ENTERED BOTH NUMBER AND NAME.

TYPE ONE OF THE FOLLOWING-

- 1 FOR RESULTS SORTED BY NUMBER WITH NAME INCLUDED.
- 2 FOR RESULTS SORTED BY NUMBER WITH NAME NOT INCLUDED.
- 3 FOR RESULTS SORTED BY NAME WITH NUMBER INCLUDED.
- 4 FOR RESULTS SORTED BY NAME WITH NUMBER NOT INCLUDED.

?3

WOULD YOU LIKE SCORES ONLY (TYPE 1),
OR SCORES AND RESPONSES GIVEN TO QUESTIONS MISSED (TYPE 2).
?2

SORTING...

CAMPBELL JEAN 10
QUESTION GIVEN ANSWER
3 D
TOTAL NUMBER INCORRECT IS 1

CASEY KARE 30
QUESTION GIVEN ANSWER
2 FALSE
3 B
TOTAL NUMBER INCORRECT IS 2

KRAM ROBE 20
QUESTION GIVEN ANSWER
2 FALSE
TOTAL NUMBER INCORRECT IS 1

QUIZ WAS TAKEN 3 TIME(S).

QUESTION*CORRECT ANSWER	INCORRECT ANSWER GIVEN FALSE	TIMES*TOTAL GIVEN*TIMES MISSED
2 TRUE		2 2
3 A	B D	1 1 2

FIGURE 3

The student's responses are stored in a file that is available for the course committee to review at appropriate intervals. The summaries provided include information on individual student progress as well as class performance (fig. 3).

The resulting system has been used in several modes, two of which are illustrated in the following section. The ability of the program to adapt to each of these modalities is an indication of its flexibility, a characteristic that has already led to its adoption by a number of other courses in various departments on the Davis campus.

Animal Physiology 110: A Comprehensive Upper Division Course

The department offers an advanced undergraduate class in systemic physiology to all students wishing to enroll. This policy, together with the success of the course itself has led to an increase in enrollment from 215 in 1970 to 304 students in the fall of 1971. The course is given through a lecture series supported by eight associated laboratory sections of 24 students each and 34 tutorial sessions with from three to twelve students.

The concept of team teaching has been thoroughly embraced by the instructors of this class. Course committee meetings are held weekly to discuss progress, review specific problem areas and outline material to be presented in the following week. Tutors, most of whom are graduate students in physiology, participate in these discussions, representing "their" class and forwarding constructive suggestions for improvements in the course. The responsibilities for disseminating information and monitoring student progress are thus shared by faculty, tutors, auxiliary audiovisual aids, and - the computer.

The role of self-evaluation in this course is illustrated in Fig. 4. Optional problem sets are handed to the student at the beginning of each week. The problem sets are written to increase the student's understanding of designated course material, identify his particular weaknesses and direct him to further sources for review. These problem sets are thus not tests but learning aids, used to prepare the student for lecture material that will be presented subsequently. When the student has completed his review of the problem sets, he enters his answers through a terminal. The computer scores his work, noting the incorrect responses and directing him to appropriate remedial information. The student may then, by finding the correct responses and discussing his revisions with his tutor, receive full credit for the entire problem set. A weighted point system provides incentive for the poorer student by offering him more bonus points for successfully completing the problem set.

While the student is completing his review, the tutors are also able to review the collective performance of their students, using the summary program described above. They are thus in a position to direct the tutorial discussions to areas most pertinent to their class. They are aided in choosing the appropriate methodological approach for their discussions by the weekly meetings of the course committee, during which the students' problems are reviewed and the appropriate points for discussion are considered. The task of clarifying difficult concepts is made somewhat easier for the tutors by limiting tutorial discussions to one week's material. The tutors find that this approach offers them a challenge that they can meet, and also an effective means for improving their own depth of understanding of some of the more difficult concepts. For their part, the undergraduates use the discussion sessions to clarify individual points that remain unclear after their independent study, to participate in group discussions about physiology, and to forward information through the tutors to the instructors regarding the progress of the course.

The problem sets are intended to be introductory to the lectures. Each instructor spends from three to four hours preparing a problem set that will permit him to extend his lectures into concepts that require an understanding of basic principles. For example, it is difficult for undergraduates to appreciate the dual interpretation of certain physiological observations, such as the sensory encoding of information by the central nervous system. Since the text adopts the approach of "specific receptors", a problem set was designed to help the student pick out the arguments used by the author to support his views. The problem set, together with supplemental reading was followed by two lectures, one which adhered to the interpretation in the text, tracing specific pathways for receptors within the central nervous system. The second lecture, delivered by a different instructor, presented an alternative concept requiring more complex encoding and multiple sensory pathways as a part of the central nervous system's information processing function. Exposure of alternative concepts is greatly enhanced by the knowledge that students had already reviewed the problem sets and were thus well informed about the basic concepts prior to presentation of the more complex alternatives.

Introduction of this system came at a time when the campus computer system was just beginning to accept a full-fledged timesharing responsibility, and a series of problems accompanied the operation from the start, including sporadic malfunctions of the remote terminals, the host computer and the systems software. Despite these major inconveniences, the

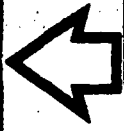


FIGURE 4. Integration of computer terminals into a large undergraduate physiology course. A student reviews background factual information (A); hears a discussion of general concepts in a large lecture (B); participates in a laboratory (C); and discusses ideas in a small tutorial session. (D)

overall reaction of the students was favorable to the system. Their enthusiasm stems partly from the manner in which this experiment was presented to them, partly from the sense of community developed by the tutorial discussion sessions, and partly because the program filled a real need identified by the student. In addition, the program designers worked throughout the quarter to modify the program in accordance with suggestions made by instructors and students. Several important changes were implemented in this fashion during the fall of 1971.

Human Physiology 298: A Specialized Graduate Course

The School of Medicine at UC Davis offers an interdisciplinary core curriculum supplemented by departmental electives. One instructor, teaching a class in renal physiology to three medical and ten graduate students, was interested in using the self-evaluation system to measure student comprehension of the lecture material already presented. Accordingly, he designed short problem sets for the students to work out within a day or two following presentation of the lecture material. The feedback received by the students was used to guide their study in a manner similar to that described above. The instructor was thus able to scan overall student performance and adjust subsequent lectures. A single terminal was available for the class during the quarter, and it was occupied only a small portion of the time by the class.

Student reaction to the system was generally favorable in this class as well. The computer aid was highly valued by the instructor. He willingly accepted the additional time spent in preparing meaningful comments to be displayed when a wrong answer was chosen in view of the time saved by having the computer score, grade, and record results of individual students.

System Implementation and Performance

The self-evaluation system was designed, coded and debugged during the summer of 1971. Several initial meetings were held to design the system early in the summer. Coding was done in ALGOL by one of the authors (GMW) as a part time activity in the latter part of the summer. The total effort involved in initial implementation coding was less than two weeks.

After the programs were introduced in the fall quarter, 1971, a number of program modifications were implemented requiring an additional week of design, coding and testing. Additional changes are planned for the winter quarter; however, the system is fully operational in its present form and will be used in several classes during the winter and spring quarters.

Future Plans

The self-evaluation system meets the criteria of providing a simple entry point into the instructional use of computers. However, this system is also designed to serve as the start of a modular development effort. Some of the plans currently under discussion are described below.

The generation of self-evaluation problem sets will be expanded to permit representative selection, with computer support, of an appropriate group of questions from a library of problems or questions that has been indexed by subject matter or complexity. Although this step would increase the cost for creating problem sets, it might greatly increase the number of separate problem groups that could be made available to students, allowing a single class to have more experiences in self-evaluation.

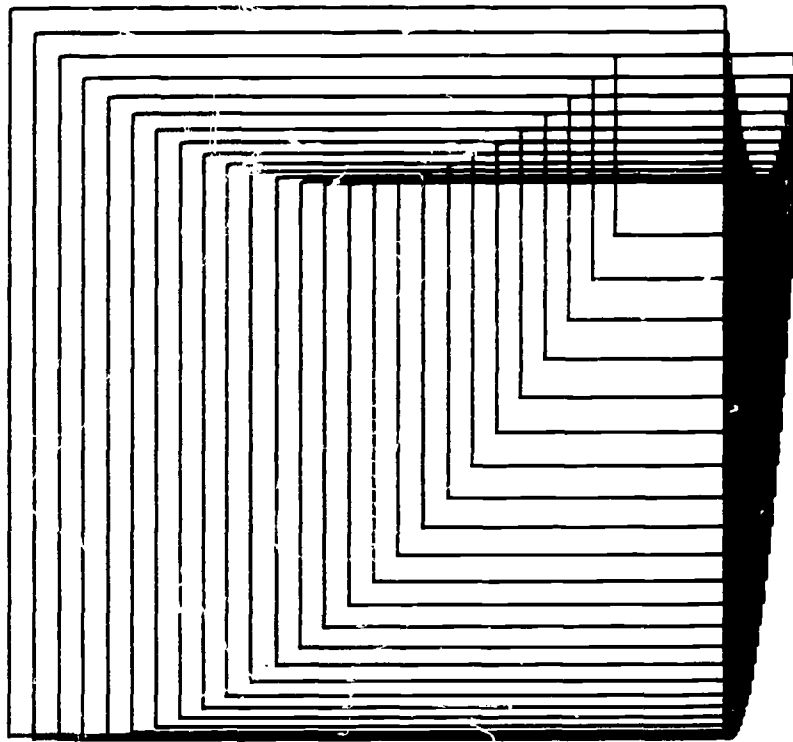
Certain primitive branching instructions will be introduced, such as skipping past questions if the performance level is adequate, repetition of certain questions following presentation of more detailed tutorial information than is currently entered, and the acceptance of more complex word answer alternatives. This approach would not develop into a complete CAI system, in that specific incorrect responses would not be anticipated. Extension to a complete tutorial language is feasible, but would be a later step in the development of this system.

The system will also be expanded by coupling it to simulation, specially designed supplemental problem sets or other related learning programs. In this method of learning, one feedback response from the self-evaluation would be to call up or recommend to the student a tutorial program available on the computer. Alternatively, a self-test problem set may be presented at the end of the auxiliary program to reinforce key concepts.

Summary

The problems of computer entry into instructional support can be overcome by simple programs with effective instructional objectives. One such program system, aimed at providing students with self-evaluation through problem sets and furnishing faculty with details of student performance, has been designed for use in both large and small classes at UC Davis. This system was designed and implemented during the summer of 1971, then introduced simultaneously in

three classes during the fall quarter. Student reactions, despite problems in the operating system, were highly favorable. The opportunities to extend the system into more complicated instructional methodologies appear promising, but the main effect has been to gain short term benefits without sacrificing either quality or long term potential. It is felt that the underlying philosophy has considerable appeal for institutions that are just beginning to consider use of computer support for instruction.



W. T. BOUTTOTE

DIMINISHING SQUARES by William (Bud) Bouttote
Problem: Diminishing squares imaginatively developed

IMPROVING LARGE ENROLLMENT UNDERGRADUATE INSTRUCTION WITH COMPUTER GENERATED, REPEATABLE TESTS

Mark Hammer and C. Obert Henderson
Washington State University
Pullman, Washington 99163
Telephone: (590) 335-3507

The Problem

Large enrollment classes are increasingly characteristic of undergraduate education, most especially for introductory, freshman-sophomore level courses. This trend toward larger lecture courses has been accelerated by recent budgetary squeezes and the resulting pressure for improved academic productivity.

When compared with small classes, large enrollment classes have several serious disadvantages. For one, they are impersonal: Socratic dialogue is impossible, classroom questions are disruptive, and personal acquaintance with instructors is discouraged. For another, they are insensitive to individual differences: large lectures must be aimed at the "average" student, with detrimental consequences for both fast and slow learners.

Perhaps the most serious of the large-enrollment disadvantages are those surrounding the examination procedures which are forced upon instructors by sheer class size. For example, essay examinations are all but precluded by the impossibility of the grading task they impose. The typical substitution of "objective" (true-false, multiple-choice) tests for essay tests tends to reduce the intellectual rigor of the course by changing the required level of learning from mastery (recall level) to familiarity (recognition level).

In addition to changing the level of learning required, the imperatives of large enrollment instruction effectively force a change in the educational role of the test itself. Tests in small classes may be utilized primarily as learning devices which provide both student and instructor with diagnostic information on the student's level of understanding. Tests in large classes, however, are harder to utilize as learning devices. The essay exams, frequent quizzes, in-class recitation, and rapid feedback which are possible in smaller classes are effectively precluded for use in large lecture sections; large-class tests are much more likely to be infrequent (two or three major tests per semester), to cover correspondingly larger blocks of subject matter, and to have longer feedback periods (if the tests are returned at all; finals frequently are not). The cumulative result is that large-class examinations are used for evaluation rather than for diagnosis, and the potential value of the test as a learning device is forfeited. The common practice of posting test grades while not returning tests themselves confirms the exclusively evaluative role of the examination process.

Finally, large-enrollment tests are likely to be aversive (anxiety-arousing; dissatisfying) to students. Several factors are responsible for this aversiveness. First, the study habits of students are commonly observed to follow a "loaf-cram" pattern, with crams coming just before tests. Second, when exams are infrequent, the subject matter to be learned during one cram is greater. Third, the "perform now or never" nature of the test situation, coupled with intense emphasis on grades, creates a high-tension situation for the student. Neither the loaf-cram study schedule nor the pre-exam anxiety are conducive to effective learning.

Computer Generated, Repeatable Testing: A Promising Development

The limitations of large-enrollment instruction have been systematically assessed by psychologist Donald Jensen, who has proposed and evaluated a variety of potential solutions (Jensen, 1966, 1968, 1969; Jensen and Prosser, 1969). The most promising of Jensen's approaches to date is computer generated, repeatable testing (CGRT).

CGRT encompasses several important changes from typical large-class testing procedures (Prosser & Jensen, 1971). First, tests are given more frequently, typically biweekly. Second, students are allowed to schedule tests at their own convenience, within broad limits. This is made possible by the provision of multiple test forms. Third, immediate feedback is provided on test performance; students are given the correct answers to test questions as soon as they have completed a test. Fourth, students can repeat tests until they earn a grade which satisfies their aspirations. Finally, testing for mastery (recall) is possible through the use of a procedure for coding responses to fill-in questions (Prosser & Jensen, 1971, p. 297).

The procedure used in CGRT to accomplish these changes is to prepare a large number of test questions for each subject matter segment of the course and to read them into a computer. The computer is programmed to generate independent test forms, each of which contains a stratified random sample of questions from the bank in computer storage. Thus, literally hundreds of tests

can be generated with no two being the same. Having pre-printed a supply of tests on the computer (in batch mode), a testing room is scheduled to be available for convenient hours during the exam week. Students may come in when they feel most ready, take an exam, get immediate feedback, and return to do additional studying if their first score is not satisfying.

Prosser and Jensen (1971, p. 304) have reported that CGRT has been successfully implemented in several institutions in a variety of subject areas including psychology, economics, accounting, chemistry, speech therapy, and English. Among the benefits said to be associated with these implementations are higher student achievement, lowered anxiety and antagonism surrounding examinations, and better attitudes generally toward both subject matter and instructors.

Implementing CGRT: Our Experience

The theory behind CGRT made sense to us, and we had heard favorable reports on the effects of repeatable testing from Jensen and others. We decided that it was worth a trial run and agreed to attempt it. Since both of us anticipated teaching one section of an introductory Personnel Administration course, we agreed to cooperate in developing CGRT for both sections. These decisions were made in the early summer of 1971, and we aimed for Fall semester 1971 implementation.

Creating the Test Bank

The first obstacle to be contended with was the required bank of test questions. Prosser and Jensen (1971) reported that the number of test questions available for any one test should exceed the number of questions on that test by six to ten times to assure adequate variation among the test forms. More recently Jensen has said that a 10 to 1 ratio is a desirable minimum (personal communication). Prosser and Jensen also noted (correctly!) that the preparation of this number of test questions is a formidable task.

Since we did not have enough time to create a complete text bank before the beginning of the fall semester, we adopted a text which had a fairly large number of accompanying objective test questions. Some of these test questions were contained in an instructor's manual and some were in a student workbook which was available to accompany the test. We adopted the workbook and included the questions from it in the question bank, thereby providing students with pre-exposure to a number of questions over the text as well as with motivation to utilize their workbooks as study aids. The task of supplementing the questions accompanying the text and of preparing questions over class lectures was divided among ourselves and a teaching assistant.

Obtaining Computer Programs

The second obstacle to be overcome in order to implement CGRT was obtaining the computer capability needed. We initially anticipated using the system developed at Indiana University by Prosser and Jensen (1971), but two problems developed. First, a telephone conversation with Jensen convinced us that it would probably take as much programming time to convert the Prosser-Jensen system to our computer (IBM 360-67) as to develop our own from scratch. Second, we had wanted to improve on the Prosser-Jensen system in several respects, the most important one being the capacity to stratify the test bank by test item type. Without such a stratification the proportion of question types on any given test could vary randomly: the number of true-false items on a given 20-question test might vary, for example, from 7 on one test to 14 on another. In the interest of achieving uniform difficulty among test forms, we felt that each form should have the same proportion of question types.

We finally decided to create our own CGRT system. Being short on both time and money, we decided to program only the test generation capability, and to postpone the mark-sense scoring and computer tallying capabilities which are part of the Prosser-Jensen system. After specifying the capacities of the program we wanted, we located a computer programmer who agreed to write the programs for \$300.00. To our programmer's credit and to our delight, the resulting programs have functioned flawlessly throughout their first semester of operation. A sample test is shown in Figure 1, which provides an idea of the format of the tests generated by these programs.

Developing Policies and Procedures

For testing purposes the 14-week semester was divided into seven two-week units, and a test scheduled for each unit. Students were allowed to take a maximum of three (later changed to four) tests during a six-day period from Wednesday of the second week of the unit through the following Monday. This testing interval covered the period from the last lecture of a unit

SAMPLE PERSONNEL TEST
FORM NO. 10

NAME
SAMPLE PERSONNEL TEST
FORM NO. 10

SAMPLE PERSONNEL TEST
FORM NO. 10

QUESTIONS

TRUE OR FALSE

RESPONSES

ANSWERS

1. A JOB EVALUATION PROGRAM CAN CONTRIBUTE TO THE IMPROVEMENT OF EMPLOYEE SAFETY WITHIN A COMPANY.	1. ITEM 6 1 1 003 ANSWER: ___	T	1. ITEM 6 1 1 003 C-S PAGE 546.
2. THE CONSTITUTIONALITY OF THE FAIR LABOR STANDARDS ACT WAS UPHELD BY THE UNITED STATES SUPREME COURT IN THE MT. CLEMENS POTTERY CASE.	2. ITEM 6 2 1 009 ANSWER: ___	F	2. ITEM 6 2 1 009 C-S PAGE 589.
3. IN MOST GROUP INCENTIVE SYSTEMS THE INCENTIVE PAYMENTS ARE NOT SOLELY BASED UPON UNITS OF PRODUCTION.	3. ITEM 6 3 1 009 ANSWER: ___	T	3. ITEM 6 3 1 009 C-S PAGE 644.
4. THE PRODUCTION LIMITS THAT A GROUP ESTABLISHES FOR ITS EMPLOYEES IS KNOWN AS A "ROGEEY."	4. ITEM 6 4 1 002 ANSWER: ___	T	4. ITEM 6 4 1 002 C-S PAGE 601.

MULTIPLE CHOICE

5. THE PROFILE METHOD OF JOB EVALUATION: A. INVOLVES THE USE OF THE MANAGEMENT GRID B. IS USED TO EVALUATE CLERICAL JOBS C. CONTAINS CERTAIN FEATURES OF THE FACTOR COMPARISON METHOD D. PERMITS THE EVALUATION OF BOTH THE EMPLOYEE AND HIS JOB	5. ITEM 6 1 2 003 ANSWER: ___	C	5. ITEM 6 1 2 003 C-S PAGE 563.
6. EFFECTIVE FEBRUARY 1, 1968, THE MINIMUM WAGE FOR THOSE WORKING IN EMPLOYMENT COVERED PREVIOUSLY BY THE FAIR LABOR STANDARDS ACT BECAME: A. \$1.70 B. \$1.40 C. \$1.50 D. \$1.80	6. ITEM 6 2 2 004 ANSWER: ___	C	6. ITEM 6 2 2 004 C-S PAGE 587.

FILL-IN

7. THE _____ SYSTEM OF JOB EVALUATION PERMITS JOBS TO BE CLASSIFIED AND GROUPED ACCORDING TO A SERIES OF PREDETERMINED WAGE CLASSES OR GRADES. THE FEDERAL CIVIL SERVICE SYSTEM IS AN EXAMPLE OF THIS TYPE.	7. ITEM 6 1 3 005 ANSWER: ___	D	7. ITEM 6 1 3 005 JOB GRADE PAGE 548.
8. WHEN A SALESMAN IS ADVANCED MONEY WHICH MUST BE REPAYED OUT OF SUBSEQUENT COMMISSIONS, HE IS SAID TO BE ON A STRAIGHT COMMISSION AUGMENTED WITH A _____.	8. ITEM 6 1 3 008 ANSWER: ___	M	8. ITEM 6 1 3 008 DRAWING ACCT. P 610.
9. THE "SLOPE" AND THE "ELEVATION" ARE TWO CHARACTERISTICS OF A _____.	9. ITEM 6 2 3 005 ANSWER: ___	V	9. ITEM 6 2 3 005 WAGE CURVE LEC 11-12.
10. UNDER THE _____ SYSTEM OF INCENTIVES, THE AMOUNT OF WAGES IS DIRECTLY PROPORTIONAL TO OUTPUT.	10. ITEM 6 3 3 002 ANSWER: ___	C	10. ITEM 6 3 3 002 PIECEWORK PAGE 604.

END OF TEST FORM NO. 10 SAMPLE PERSONNEL TEST

FIGURE 1.

Percent (Number) Responding

CGRT Rating	Much Worse	Considerably Worse	Slightly Worse	Average	Slightly Better	Considerably Better	Much Better
After 2nd CGRT Test (N=71)	7.0 (5)	5.6 (5)	7.0 (5)	7.0 (5)	25.4 (18)	36.6 (26)	11.3 (8)
After 6th CGRT Test (N=64)	1.6 (1)	3.1 (2)	7.8 (5)	9.4 (6)	31.2 (20)	28.2 (18)	18.8 (12)

TABLE 1. Student Ratings of CGRT v. Conventional Tests

Number of Students (N=81)

Grade	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Six-Test Average
A	25	30	38	46	36	36	13
B	29	26	26	26	26	22	40
C	13	12	12	6	11	16	20
D	9	9	3	2	6	4	5
F	5	4	2	1	2	3	3

TABLE 2. CGRT Grade Distributions for Six Biweekly Tests

292

until the first lecture of the next unit (students had two lectures and one small discussion group weekly).

A testing room was manned by an instructor or an assistant for six different scheduled periods, including one period Saturday evening and another Sunday evening. Testing room procedure called for a student to sign for a test in a log book and to indicate there his discussion section and the form number of the test he received. Upon completing the test, the student would cut the "Responses" column from the test questions (Figure 1) with a pair of scissors provided, and hand it to the instructor on duty. The instructor would take the "Answers" column, which had been previously cut off, line up the correct answers with the student's responses, and grade the student's test. This grade was then marked on both the "Responses" column, which was kept for recording, and on the "Answers" column, which was returned to the student.

Having agreed that an arbitrary, pre-established criterion schedule for grading was preferable to the use of grade "curving" we adopted a fairly exacting standard, viz., 95%+ = A, 90%+ = B, 85%+ = C, 80%+ = D, and below 80% = F. We assured ourselves that students could be expected to attain levels higher than are typically demanded because: a) some of the test questions used were taken from their workbook, giving them a pre-exposure to some items, b) up to half of the test questions were of the True-False type, and c) any chance variation in test difficulty worked in the students' favor since only the highest test score was counted. Even with these considerations the grading standards seemed to us plenty rigorous, but we reasoned that we could be lenient in final grading if they turned out to be too demanding.

Results

Student Attitudes

Twice during the semester feedback was solicited from students on several aspects of CGRT. The first set of student ratings was obtained in the fifth week of the semester, which was just after the second CGRT unit test; the second set was gathered in the thirteenth week, after the sixth test. Both sets asked for open-ended comments on several specific goals and mechanics of the CGRT technique, as well as an overall evaluation of CGRT.

The open-ended responses were favorable overall, with two exceptions. Specifically, the responding students practically all had favorable responses to inquiries on fairness in evaluation and grading, repeatability, frequency, student-scheduling, and availability of immediate feedback on performance. There was also substantial agreement on two criticisms of our CGRT program: test unreliability, and excessively high grading standards. Both of these criticisms will be discussed below.

For the overall evaluation, students were asked on both occasions to rate CGRT "in comparison with other testing procedures you have seen" on a 7-point scale from "much worse" to "much better." The student responses are summarized in Table 1. On the average, students rated CGRT "slightly better" on both occasions (mean scores were 5.0 and 4.8 respectively). However, Table 1 shows that the distribution of ratings shifted from the first to the second evaluation; while the modal response decreased from "considerably better" to "slightly better," the number of "much worse" and "considerably worse" ratings decreased and that of "much better" ratings increased. Additionally, students were asked on the second evaluation occasion to indicate whether they would choose a class with a) CGRT or b) Conventional testing, if all other things were equal. Of 64 answering students, 49 (or 77%) chose CGRT.

Test Performance

Student performance on tests has exceeded our expectations. Table 2 shows the grade distribution for each of six tests that have been administered to date, as well as for the six-test average grades. There seems to be a general trend toward higher grades, and after six tests the distribution of average grades is skewed upward with a distinct mode at the "B" grade level.

Our initial expectation was that our achievement standards might have been too high. Our early doubts were amplified by student responses on the first questionnaire; many students complained that our standards were too high and unrealistic. However, after six tests, almost two-thirds of the students have averages of "B" or better. It appears to us that the distribution of final grades will be higher than the distributions either of us has seen recently in this course.

However, our standards may be too high. It is quite clear to us that the higher grades reflect a considerably higher level of effort on the students' part. We asked students on the first questionnaire how much time they were spending on this course, and how this time compared with that spent on other courses. Of the 60 responding, 2 claimed they were spending less time compared with 54 who reported they were spending more time. Whether or not it is legitimate to

utilize techniques which effectively extort a disproportionate amount of the student's study time is a question with which we have only skirmished, but which appears likely to be controversial.

We had expected some expression of resentment on the questionnaires over the increased study time which students were devoting to the course. To our surprise, the students generally expressed gratitude for being allowed the opportunity to improve their scores by repeating tests. Given the overall favorability of sentiments expressed and the pattern of test performance observed, it seems clear to us that our students are both learning more and liking it better!

In addition to improvements in performance and attitude, several other phenomena associated with CGRT deserve comment. First, it is quite evident to us that CGRT has eliminated a great deal of the aversiveness normally associated with the testing experience. Students come to the testing room relaxed and, occasionally, in a playful mood. They frequently ask questions both before and after taking their test, and the most common response to having their tests graded is to grab their text and check on incorrect responses. In short, tests are really functioning as learning devices which stimulate further study.

A second phenomenon associated with CGRT concerns student attitudes toward the instructors, who are increasingly being viewed in a coaching role, rather than in an adversary role. Having shared the task of test item construction, and having settled on a fixed and exacting set of standards for grades, we are more prone to honestly encourage each student to do his best. When a student does well, we are elated along with him. When one does poorly, his disappointment is also ours. The students seem to sense that we are really on their side and appear more prone to relate to us as helpers.

A final phenomenon associated with CGRT is that students are becoming aware that they have direct and immediate control over their own grades. When this realization is coupled with the opportunity to repeat tests until a satisfactory grade is earned, the effect is that the student's ability to rationalize a poor test score is eliminated. We emphasize this point because we think that it may be one of the most important observations to be made in connection with our experience.

To illustrate: we suspect that a substantial number of college students having actual grade point averages of "C" or lower really prefer to think of themselves as "A" or "B" students. Professors who have observed closely the typical post-examination behavior of students will agree, however, that inferior performance doesn't necessarily threaten one's self-image. Why? Because there are so many good, plausible explanations for poor performance: "Misleading test questions," "incompetent instructor," "lousy text," "testing room too hot," "headache (didn't sleep a_l last night)," "my great aunt died," "my girl left me and I'm all messed up." These familiar rationalizations (and countless others) are all invoked by students to effectively convince themselves and others that they are really better students than the record indicates.

None of this nonsense is effective under CGRT, and we think that this may explain much of the increasing scarcity of C's, D's, and F's in our grade distributions. Interestingly enough, a number of best students have shown signs of the same effect. Some seem quite incapable of settling for anything less than a perfect score. For example, students who have earned an A (19 of 20 correct) frequently return a second and a third time in attempts to make the perfect score.

Test Reliability

It was mentioned earlier that test reliability was the subject of considerable student criticism. It seemed that students all too often received lower test scores in spite of greater preparation. Our perusal of the patterns of test scores confirmed that there was at least some problem, since there were occasional instances in which a student would get, for example, a B on the first test followed by an F on the second. We were therefore led to investigate the reliability problem further.

We did a check on the test-retest reliability of one of the CGRT unit tests using four different groups of students. For a class of 112 freshman and sophomore Introduction to Business students, the reliability coefficient was .25. Reliability for a group of 23 advanced personnel students was only slightly better at .38. The highest reliability was obtained with a group of 14 MBA students, where the figure was .61. Finally, students in the present CGRT course were given two tests during one class period (both for credit), and the resulting reliability figure for these 76 students was .46.

These reliability figures were disappointingly low. The students were all too right - apparently the process of randomly selecting test questions from an item bank results in a wider variation in overall test difficulty level than we had anticipated. As a result of this information we have been thinking about ways to improve test reliability. The most promising

approach now seems to us to be that of stratifying the question bank by concept, rather than by textbook chapter or by time period (e.g., Week 8). This procedure would have the effect of reducing the variance in test difficulty attributable to variance in topical coverage. We are beginning to think more in terms of clusters of fairly equivalent questions being associated with each key objective or concept to be covered. Of course, a second sure-fire way to improve reliability is to increase the test length; so far our tests have had 20 questions each. Whether or not the increased reliability of a 30-question test would offset the disadvantages of the longer test is not yet clear.

Possible Implications of CGRT

The following speculations are offered to suggest the range of potential impact possible if CGRT proves successful.

1. Many large-enrollment, introductory courses have multiple sections and multiple instructors, and it is no secret, among students at least, that substantial differences exist among sections which are attributable to different instructors. It seems to us that there are too many instances of multiple-section introductory courses where substantial differences in course content exist. Where a certain course is a prerequisite to others, or is required for a major, substantial differences among sections of multi-section courses cause untold problems for instructors of advanced courses and student advisors. Clearly, standardization of courses at the introductory level is needed.

The possibility of cooperation among instructors for the purpose of developing a test item pool for a course suggests cooperation in defining the goals for the course. It seems plausible, if not likely, that instructors should be able to reconcile whatever differences exist among themselves and agree on specific course goals and the associated test pool questions and criteria for satisfactory performance.

One interesting question suggested by the above is "What would happen if a department were to require that instructors assigned to a certain multi-section introductory course participate in establishing a mutually acceptable set of course objectives, a test item pool, and the level of satisfactory performance?" Surely some groups of instructors could do this with little inconvenience; almost as surely as some could not. However, it may be that those instances where irreconcilable differences exist are precisely those where departmental-level intervention is appropriately exercised to eliminate minority individuals or factions from teaching the introductory course. This may sound severe, but it boils down to the reasonable proposition that introductory courses should concern themselves with consensus-level subject matter.

This should not be taken to imply that the course in question should be highly structured in either content or method; one group of instructors might, for example, decide that their "consensus topics" should constitute 25% of the course requirements, and the remaining 75% would be open to the individual instructor's preference. Furthermore, the methods used by the instructor to cover the consensus topics would be quite open.

2. If the development of consensus-level test item pools is a practicable possibility, and these were to become available for major undergraduate courses, a number of interesting advantages might be realized. For example, take a transfer student who had taken an introductory math course at another institution: he is satisfactorily prepared to begin work in advanced courses? The availability of consensus test pool would make it possible to give the student a subject matter mastery test which would pinpoint any areas of weakness.

Such tests might be useful in determining whether students should be given credit for various combinations of prior work. The effect of such a practice might well be to shift the criteria for acceptability from such arbitrary consideration as, "Was his institution accredited?" or "What text did he use?" or "Where and when did he take the course?" to "Does he now understand the critical concepts?"

3. Another major advantage of the existence of CGRT tests would be that superior students could be invited and challenged to proceed at their own pace and to demonstrate their competence as soon as they are ready.

4. A further implication of the widespread availability of CGRT test item banks is that independent and off-campus study could be greatly facilitated. If course objectives and requirements were specified and made available along with sample CGRT tests, all eligible applicants could be invited to demonstrate their competence on any available CGRT test, and to claim credit and advanced standing for doing so.

Incidentally, CGRT tests would seem to be ideally suited to correspondence study. For one thing, numerous sample tests could be provided to the correspondent-student. For a second, the immediate feedback on test performance possible with CGRT would be a dramatic improvement over

the long-delayed feedback typical of correspondence course tests. Finally, the use of the same CGRT exams being used in parallel courses on campus would insure the comparability of the two courses in subject matter coverage.

5. CGRT appears to be highly compatible with several concepts associated with the audio-tutorial approach to learning (Postlethwait, Novak & Murray, 1969). Student scheduling, repeatability, and prompt feedback from frequent quizzes are features of both. The concept of providing mini-courses and requiring learning for mastery (Bloom, 1969) suggests that CGRT test pools could be geared to mini-courses and the criterion level stated. Furthermore, specifying objectives in behavioral terms (Mager, 1962) is a step which should naturally precede preparation of the specific test bank items which operationalize those objectives.

Further Development Anticipated

CGRT has been surprisingly successful, and we have plans for expanded implementation and for more systematic experimental evaluation. Mark Hammer has recently received a \$12,000 grant to develop a more sophisticated CGRT computer system and to give CGRT a more thorough evaluation compared with conventional testing techniques. As one result of this project, computer programs and documentation should be available by September 1972.

REFERENCES

1. Bloom, B. S., Learning for mastery, UCLA CSEIP Evaluation Comment, 1968, Vol. L, No. 2.
2. Campbell, Donald T., and Julian C. Stanley, Experimental and quasi-experimental design for research in teaching. In N. C. Gage (ed.), Handbook of Research in Teaching. Chicago: Rand McNally, 1963.
3. Hammer, Mark A., C. Obert Henderson, and LeRoy Johnson, A report on four techniques for improving large enrollment instruction. Unpublished summary of report given at Northwest Universities Business Administration Conference, Portland, October 30, 1971.
4. Jensen, D. D., Testing and large class instruction. Growth and Change at Indiana University, 1966, Vol. III, Section XLI.
5. Jensen, D. D., A system for improved large enrollment instruction. Unpublished manuscript, Indiana University, 1968.
6. Jensen, D. D., An efficient, effective, and humane system of large class instruction. Unpublished manuscript, Indiana University, 1969.
7. Jensen, D. D., and F. Prosser, Computer-generated, repeatable examinations and large class instruction. Paper presented at Midwest Psychological Association Meeting, Chicago, 1969.
8. Mager, Robert F., Preparing Instructional Objectives. Palo Alto, Calif.: Fearon, 1962.
9. Postlethwait, S. N., J. Novak, and H. T. Murray, Jr., The Audio-Tutorial Approach to Learning, 2nd. ed., Minneapolis: Burgess Publishing Co., 1969.
10. Prosser, Franklin, and D. D. Jensen, Computer generated repeatable tests. AFIPS (American Federation of Information Processing Societies) Conference Proceedings, 1971, 38, pp. 295-301.

AN ANALYSIS OF THE USE AND EFFECTIVENESS OF EXAMINER - A COMPUTERIZED QUESTION BANK
AND EXAMINATION PROCESSING SYSTEM - IN COLLEGE OF BUSINESS COURSES AT THE UNIVERSITY OF
SOUTH FLORIDA

Stanley J. Birkin
University of South Florida
Tampa, Florida
Telephone: (813) 974-2960

Introduction

A considerable number of the undergraduate "core" courses in the College of Business at the University of South Florida are taught to large classes with as many as 250 students in a section. Such an arrangement presents the instructor with enormous problems concerned with the presentation of the course material itself and also with the administration of the course. These administrative problems include: (1) examination preparation and administration; (2) class handout preparation, and distribution; (3) grading of examinations and various handed-in assignments; (4) providing the student with feedback as to "how he stands" in the course; and (5) determination of the final course grades based on overall performance. This paper deals with a computerized approach designed to alleviate the first of these administrative problems namely that of preparing and administering the examinations in the course. The paper reports on the use and effectiveness of the EXAMINER system which was designed and made operational by the author. The system has been used in the preparation of examinations for the PRINCIPLES OF MANAGEMENT courses and the COMPUTERS IN BUSINESS courses over the past four academic quarters.

The Need for Improvement

The impetus to improve upon previous methods of examination preparation and administration came from three directions. The first was that the sheer size of the classes (in terms of numbers of students) meant that the examinations were typically of an objective nature and involved multiple-choice and/or true-false questions. Each time the course was taught the instructor had to, (a) compose new questions, or (b) draw upon questions from previous examinations, or (c) use questions from the instructor's manual or some other source. An informal survey showed that many instructors kept a card index file containing their "favorite" list of "tried and true" questions. Selections of questions from these files, coupled with additional questions from elsewhere were used each time an examination was needed. This involved considerable amounts of retyping of questions each time an examination was prepared. The second was that large classes are often conducted in either a room or auditorium in which the students sit close together. Consequently, there is always the opportunity for a student to observe the responses marked by nearby classmates on their answer sheets. The third direction was that the demands for secretarial work in the department were such that examination preparation involving typing, duplicating and collation had to be minimized.

The EXAMINER System

EXAMINER is a two-phased system. Phase I involves the establishment and maintenance of a computerized question bank. Questions are punched on to cards in a fairly free format style and become part of a permanent card, tape or disk file. This provides a readily accessible means of retrieving questions for processing by the second part of the system. Phase II is the examination processing part of the system and operates as follows. On request, the EXAMINER system produces for the instructor in normal examination format a numbered listing of all or some of the questions in the bank. From this list, the instructor selects (along with an additional group of questions not currently in the bank) those questions he wishes to include in the examination. These question numbers are punched into predetermined fields on standard 80 column cards. Required headings for the examination and also instructions to be included at the beginning of the examination are punched on to standard 80 column cards. An EXAMINER control card is punched which provides information such as:

- a. How many copies are needed?
- b. Is the examination to be produced in two parts with a certain group of questions in PART I and a second group in PART II?
- c. Should the examination be produced in a single-print format (in columns 1-64 of the paper output) or double-print format (with two examinations printed side-by-side on the paper output in columns 1-65 and 68-132)?

- d. How many different times should the questions be scrambled? The system provides for up to a maximum of four different scrambled styles of presenting the same question.

A complete listing of the computer system job control language (JCL) cards, and the EXAMINER control and instruction cards for a sample run is given in Appendix A. The output produced by this sample run is shown in Appendix B. A sample answer key produced by EXAMINER is presented in Appendix C.

The physical reproduction of examinations by EXAMINER can be performed in one of two ways. The first way, and that used at present at the University of South Florida, is to request the EXAMINER system to print sufficient copies for the entire class. Using multiple-part paper and the side-by-side, double print format feature 200 copies of a twenty page examination typically has taken approximately 20 minutes on the high speed printer. The advantage of this method is that after removing the carbon paper and bursting the multiple copies, the examinations are ready for stapling together and do not have to be collated page-by-page manually. The second way is that the EXAMINER system can be used to produce a single master from which the examination can be reproduced by conventional duplicating methods.

Student Reactions to EXAMINER

A survey of over 200 students enrolled in courses using EXAMINER gave some indications of student reactions to the system. The students were asked to respond to four forced-choice questions giving their opinions concerning EXAMINER. The results of the survey were as follows.

Question 1. When you took your first examination using EXAMINER did you find that its "computer format" appearance was a distraction? The survey clearly indicated that the initial encounter with EXAMINER causes some distraction because of the appearance of the examination.

<u>Response to Question 1</u>	<u>Number</u>	<u>Percentage</u>
Definitely	20	9.4
To an extent	21	9.9
Only a little	58	27.2
Not at all	83	39.0
No, it was a help	31	14.5
Total Responses	212	100.0

Just under half of the students responding (46.5 per cent) believed they were distracted to varying degrees by the appearance of the examination. However, 39.0 per cent reported no distraction at all, and 14.5 per cent of the respondents indicated that the clarity of the EXAMINER format was a help in taking the examination.

Question 2. In subsequent examinations that you took which were prepared by the EXAMINER system, did you find that this level of distraction (if one existed) diminished?

<u>Response to Question 2</u>	<u>Number</u>	<u>Percentage</u>
Yes	80	37.7
Stayed the same	99	46.7
No, it increased	33	15.6
Total Responses	212	100.0

This question was designed to determine if any distraction caused by examinations produced by the EXAMINER system decreased as the students became familiar with computer produced examinations. As the responses show, over a third of the students in the sample 37.7 per cent believed this to be so. Also 46.7 per cent reported no change in the distraction level as subsequent examinations were taken.

Question 3. As far as the quality of the reproduction of the examinations produced by EXAMINER is concerned did you find that they were legible?

A follow-up on a sample of the 31 responses indicating that the examinations were poorly reproduced showed two sources of concern. The first was that the carbon copies produced using

five-part paper were faint and hard to read in a large auditorium with only limited lighting facilities. The second was the use of all upper case characters in the printing of the examination. The first of these problems has already been remedied by the

<u>Response to Question 3</u>	<u>Number</u>	<u>Percentage</u>
Definitely so	51	23.7
Reasonably so	133	61.9
No, poor reproduction	31	14.4
Total Responses	215	100.0

use of three-part paper with good results. The second problem has not been resolved at this time. An extension of the EXAMINER system to incorporate upper and lower case characters using special print train on the printer would provide one solution to the problem.

Question 4. What is your overall reaction to taking examinations produced by EXAMINER?

<u>Response to Question 4</u>	<u>Number</u>	<u>Percentage</u>
Favorable	66	30.7
Neutral	99	46.1
Unfavorable	50	23.2
Total Responses	215	100.0

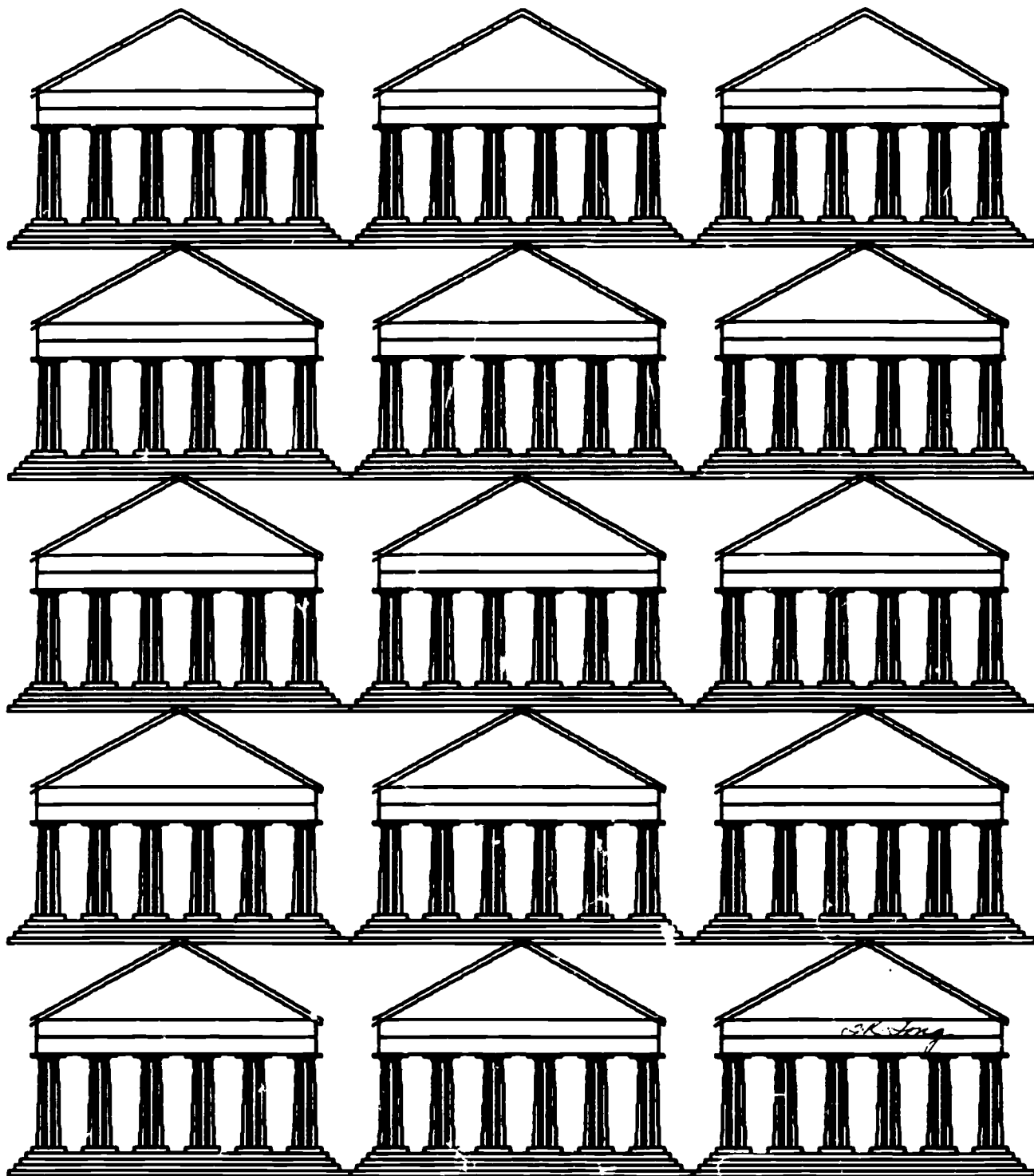
Since the EXAMINER system is an administrative program designed primarily to benefit the instructor rather than the student, the percentage of favorable and neutral responses to this question is regarded as highly encouraging.

Summary

The EXAMINER system can be an extremely valuable approach to improving the administrative effectiveness of the examination preparation process. The major benefits result from four features of the system.

1. The maintenance of a permanent question bank for a given course, which provides an easy method of preparing an examination by selecting questions from the question bank.
2. The reduction in the amount of secretarial typing, reproducing and collating of examinations.
3. A reduction in the opportunities for cheating of scrambled styles of producing the same examination.

At this time details of the cost effectiveness of the EXAMINER system are not available. A study of the comparison of the cost of computer versus manually produced examinations is underway. It is hoped that such a study will reveal even more additional long-range benefits of the EXAMINER approach to computerized examination processing and question banks.



PARTHENON by Tom Fong

Problem: Redundant serial imagery

A COMPUTER-ASSISTED METHOD FOR TEACHING LARGE ENROLLMENT LECTURE
SECTIONS: THE BIOLOGY PHASE ACHIEVEMENT SYSTEM (PAS)

R. G. Flanke, W. D. Dolphin, G. P. Covert
Iowa State University
Ames, Iowa

C. D. Jorgensen
Brigham Young University
Provo, Utah

Introduction

Iowa State University initiated an interdisciplinary undergraduate program in biology at the beginning of the academic year, 1969-70. This new program substituted several new biology courses at the freshman level for redundant introductory courses in botany and zoology. One of the new offerings was a lecture course, Principles of Biology, often taken as the first course in biology at the University by majors and non-majors.

Principles of Biology grew quickly. Lecture sections of 300 to 600 students were created to accommodate the large enrollment, a total of 3541 students in the 1969-70 academic year, and 3571 students in 1970-71.

The shortcomings of the teaching and administrative procedures followed in this and many other large lecture courses soon became obvious. As a result improved methods were sought which would meet the problems in biology and possibly be applicable to large courses in other disciplines too.

Teaching/learning inadequacies in these sections included the following:

1. the lack of student individuality in the learning situation. Highly motivated or well-prepared students were often bored with the material presented. Ill-prepared or slow-learning students were too often overwhelmed. Consequently, a method was desired which would allow students to set their learning pace in accordance with their background and ability.
2. the dependence of the student on the lecturer and textbook for his biological information. Instructors felt strongly that learning should primarily be an enterprise of the student and that the purpose of the lecturer was to guide the student's search for information rather than primarily to provide biological facts and principles. A method was needed which would encourage the student to use the instructor, the book and other facilities as resources to be consulted for answers to specific questions.
3. the competitive grade-oriented structure of the college biology experience. The most valuable expenditure of time and energy for students was felt to be the individual's increase of knowledge over his previous level of learning. Therefore, a system was desired wherein the student would compete in learning with himself and not with others.

Administrative problems included:

1. record keeping. Most instructors usually administered three exams per term. Grades from these exams were totalled, "curved", and letter grades assigned. Consequently, much time was spent on record keeping rather than teaching.
2. the development of exams. Exam production took many hours. Numerous forms were required because of the crowded lecture halls, and usually elaborate precautions had to be taken to insure security.
3. the lack of any systematic and easy assessment of the new program. The young program was felt to need constant evaluation so as to know where strengths and weaknesses lay. Furthermore, large numbers of students in the basic program seemed to present an ideal situation for obtaining data on basic teaching/learning problems. A means had to be created to secure data systematically and easily.

An attempt to meet these problems resulted in a system of teaching large groups in the basic biology course, Principles of Biology, using computer assistance. The system, known as the Phase Achievement System (PAS), was developed jointly by the Biology Program and the Computer Center at Iowa State University using the Test Scoring Service facilities.

The Method of PA.

In order to implement PAS, the course content was modularized and computer programs were developed to handle the record-keeping and the exam compilation and evaluation.

Course Content

The course content of Principles of Biology was arranged into eight areas, each centered around a conceptual theme. These eight areas, or Phases, were

1. Structure and Evolution of the Cell
2. Cellular Metabolism: Catabolic
3. Cellular Metabolism: Anabolic
4. Mitotic Cell Division and the Development of Multicellularity
5. Reproduction
6. Genetics: Inheritance
7. Genetics: Molecular
8. Evolution

For each phase a set of behavioral objectives was constructed. This was a time-consuming project forcing extensive examination of the definition and purpose of Principles of Biology by individual instructors. After the course objectives were formalized the students were given copies and told that these represented the material of the course. The student was then free to use whatever resources the University had to meet these objectives.

Multiple choice examination questions correlated with the behavioral objectives were written and collected for each phase. Eventually over 3,000 questions were accumulated for all phases. Besides being grouped by phases, questions were subgrouped into 10 topic categories within a phase and difficulty estimates assigned to each question on a 1-10 scale of decreasing difficulty. Eventually difficulty estimates will be objectively assigned on the basis of the percentage of the students answering the question correctly.

With the modularization and definition of course content and the adoption of a standard pool of test questions the following procedures were adopted:

1. Lecture schedule and detailed course objectives were distributed to students.
2. At the beginning of the term a short test over each phase, a "Test Out exam," was administered. Each phase test contained 15 questions, the exam containing 120 questions. Any student who passed seven out of eight tests could at this point receive course credit if he desired. Students who passed less than seven out of eight tests attend lectures in order to prepare themselves for the next exam. By the end of the quarter students had to accumulate passing grades on seven out of eight phases in order to receive a passing grade.
3. At each two-week period throughout the term an opportunity was provided for the student to re-take an exam over any phase not previously passed, or passed with a low mark. Thus, a student was provided with the opportunity to re-take any phase exam in an attempt to pass a phase or raise a grade on a phase exam already taken. In a ten-week term a student had the opportunity to take an exam six times over each of the eight phases. The plan allowed a student to arrange his own exam schedule.
4. Students' grades were assigned only when passing scores were attained on seven of the eight phases. A score of 53% was "passing" on each phase. Students who completed at least four of the seven phases, and who took exams over other phases but did not pass them, were eligible for a grade of Incomplete (I). To receive this grade the student signed a written agreement accepting an Incomplete. Any student receiving an Incomplete was allowed another term to pass exams and bring his record up to passing seven out of eight phases. These procedures are illustrated in a generalized flow chart in Figure 1A.

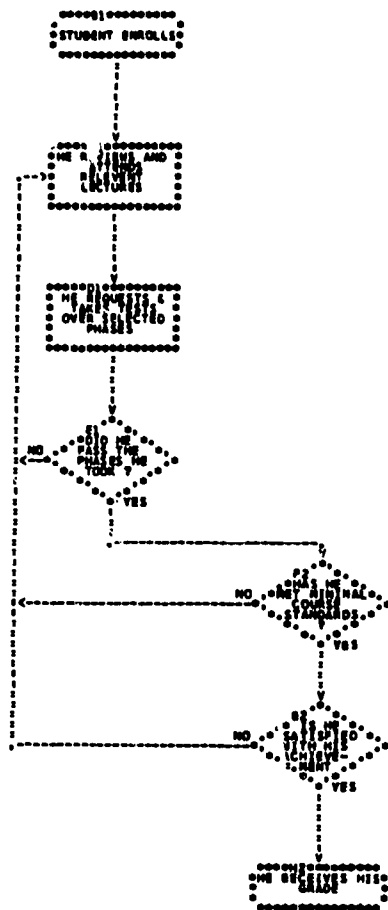
Computer Assistance

Two computer processes have been developed to support the curricular innovations. These are the scorer and record keeping process, and the test generator process. Descriptions of these processes from a usage point are detailed below.

ASPECTS OF THE PHASE ACHIEVEMENT SYSTEM (PAS)
 OF THE BIOLOGY PROGRAM OF THE COLLEGE OF SCIENCES AND HUMANITIES
 AT IOWA STATE UNIVERSITY AMES, IOWA

FIG 1-A

PAS FROM STUDENT'S VIEWPOINT



THE GENERATOR AND SCORING PROGRAMS
 ARE RUN ON THE IOWA STATE UNIVERSITY
 130 COMPUTATION CENTER

THE 130 STUDENT COUNSELING SERVICES
 TEST AND THE BIOLOGY PROGRAM
 USE OPTICAL READERS TO CONVERT
 ANSWERS TO A MACHINE READABLE FORM

FIG 1-B

FLOW OF PAS PROCESS

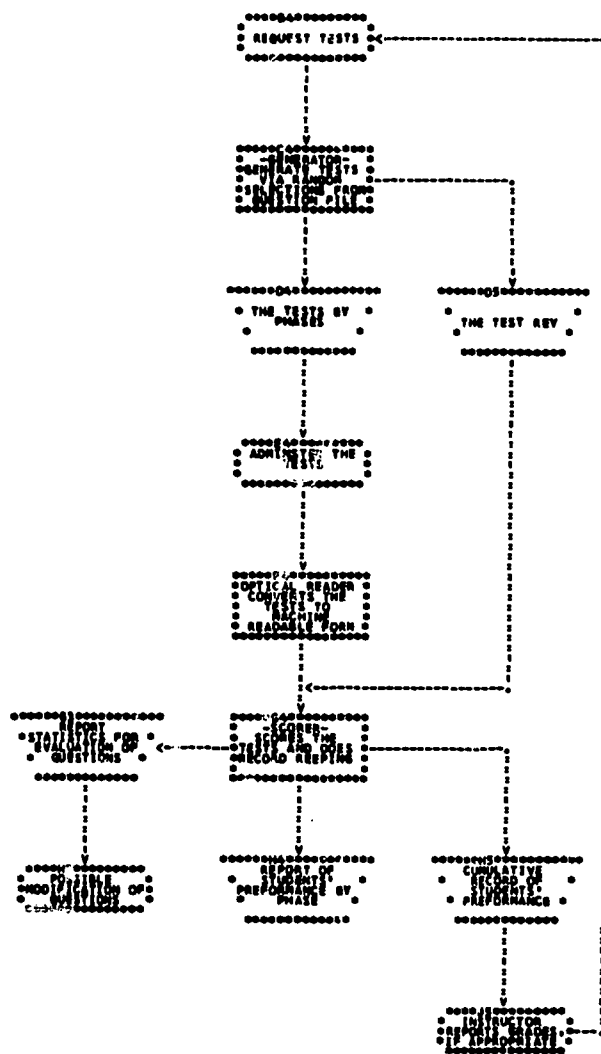


FIGURE 1. The overall flow of the Phase Achievement System.

A. Sequence of events followed by students enrolled in course.

B. Sequence of events involved in testing and scoring of students.

STUDENT RECORDS FOR BIOL 101 PAS EXAMS																	DATE:09/NOV/71								
SSN	ZE	DT	NP	SC	PHZ	1	PHZ	2	PHZ	3	PHZ	4	PHZ	5	PHZ	6	PHZ	7	PHZ	8					
3474680665	1110	0	0	47		40		33		20	27		7	20		20	20		13	13		13	7		
3474684201	1110	4	60	73		40	53		40	20	47	33	47	60	13	53		40						47	
3475646570	1110	6	55	60		53			27	13	53	47	47	47	27	27	53	40	40	60	47	27	33	53	53
3476607307	1110	3	62	60		33		40		53		40				40	47		47	73		47	47		
3478626850	1110	8	79	80		67	80		53	80		67	93		60	80		67		67				73	

Fig. 2. An example of instructors' cumulative record of five students. Abbreviations: SSN, section number and then nine digits of social security number; ZE, Security function, a digit appearing here means student record has been altered by other than test input; NP, number of phases passed to date; SC, student's average only on those phases passed; PHZ 1-8, students scores on phase tests taken. These records are taken after third exam period. Further scores earned by students will be entered below these scores.

DATE:15/NOV/71										
SSN	NP	SCORE	1	2	3	4	5	6	7	8
3481586420	7	83	80	80	80*100	80	87*			73*
3481626696	2	60	60	33	60	40#				
3481642803	3	55	53	60	47	53	33#	33	40	40
3481649392	8	87	87	87	93	87	80	80	67	93
3481680598	7	70	67	80	67	67	87*	67	47*	53
3481686670	0	0	47	20#	27	33	40	40#		
3481700996	7	68	80	67	53	47*	73*	67	53*	80*
3481705455	7	70	67	67	87	60	47*	87*	53*	67

Fig. 3. An example of exam results as posted. Abbreviations as in Fig. 2. Entries without superscript were carried forward from previous exam. Entries with * or # represent scores from current exam. A * means current score is highest score earned to date. A # means current score is lower than a score previously earned on particular phase. Average is based on highest passing scores earned on a phase regardless of whether score is printed or not.

OVERALL FLOW OF THE PAS PROCESS

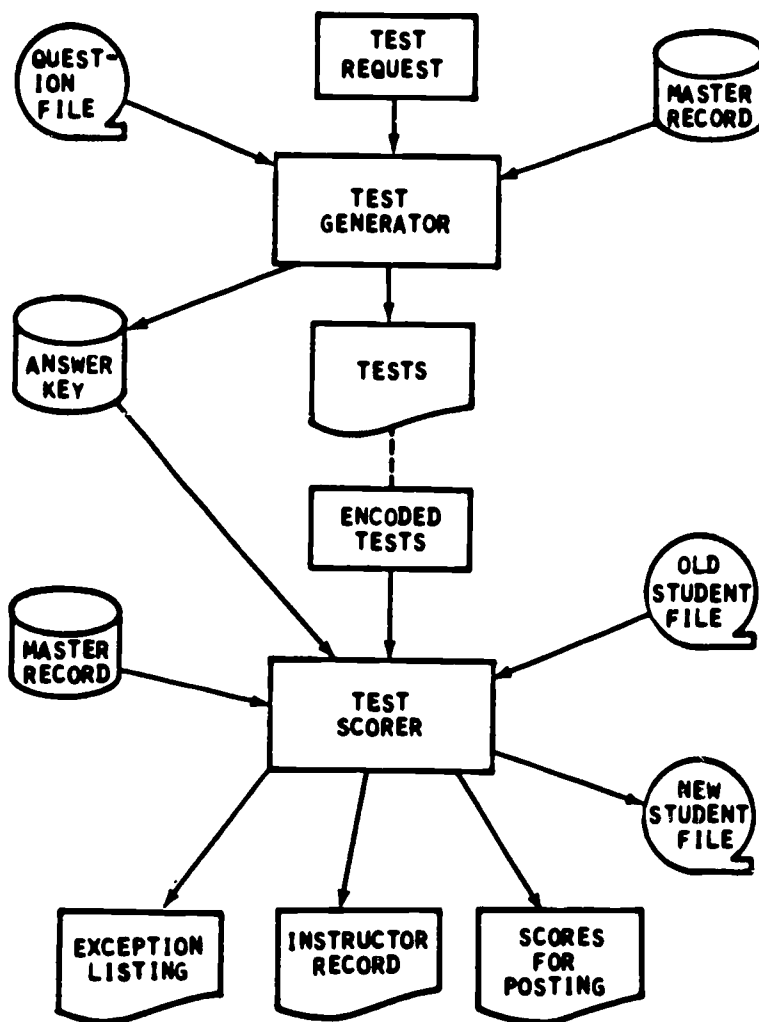


Fig. 4: Flow chart of generator and scorer programs showing inputs to various programs. Master record contains constants relating to number of phases, number of questions per phase, passing levels, relative weights for each phase (one is used for each phase in case discussed) etc. Programs are highly adaptable to other courses or changing courses with this provision.

ANALYSIS OF THE ELEMENTS OF A GRADE

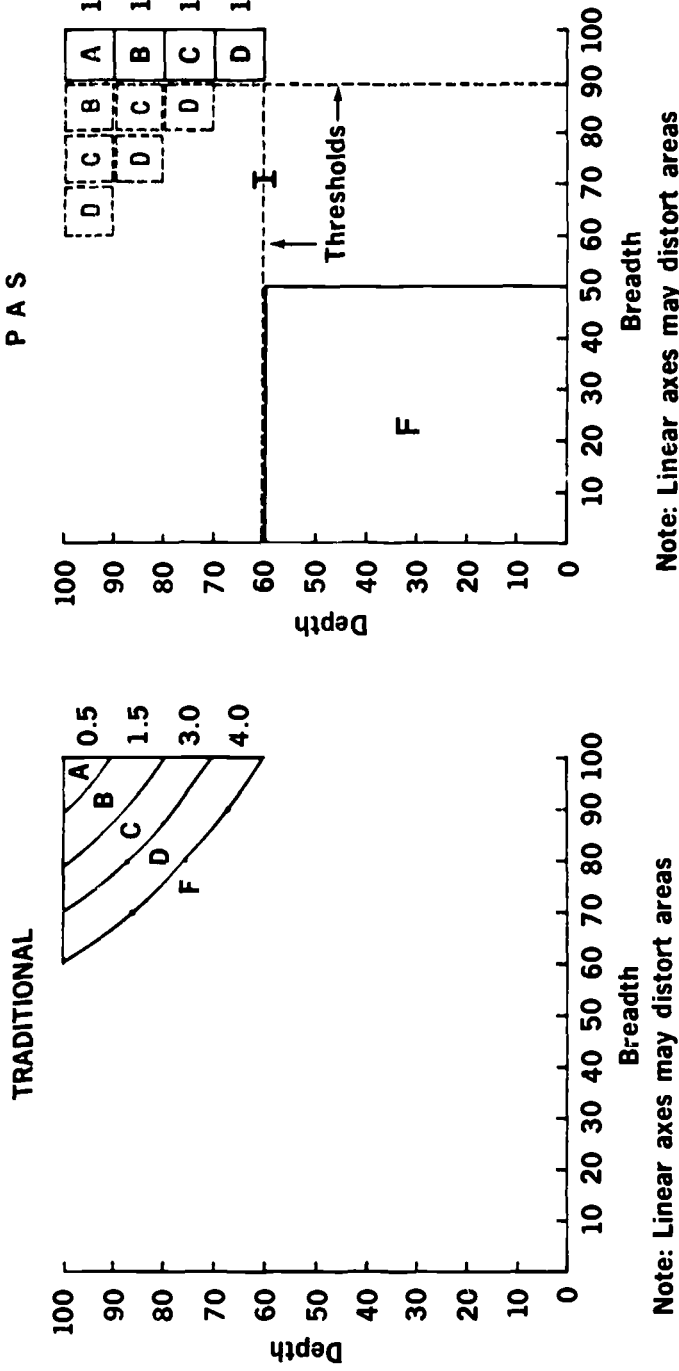


Fig. 5. A linear plot of what a grade in a traditional course means in terms of depth and breadth of student understanding. If a 60% score is considered passing, then area of performance is outlined by figure in upper right. Passing grades are set up at 60, 70, 80, 90. Numbers to right of figure give relative areas of performance for each grade. Cf. to Fig. 6.

Fig. 6. A plot of the elements involved in a grade awarded under PAS. PAS imposes a dual threshold on student. He must reach a certain level on a phase in order to pass that phase and he must pass seven phases to pass the course. Once these thresholds are exceeded his grade is determined by his depth performance alone. In this way both the breadth and depth of a student earning a grade can be specified. Students failing to pass four out of eight phases fail. Students passing four but not seven receive an incomplete grade (I). Grades enclosed in dashed lines indicate alternative grading schemes that could be used if breadth control is not utilized in passing 80% of the course with a 90% or greater average would equal B. This feature would allow the PAS grading scheme to approach the traditional (Fig. 5). Numbers to the right of the figure indicate relative areas of performance for passing grades under PAS.

1. The Scorer and Record Keeping Process. Students taking an examination pencil mark standard IBM Form 511 multiple choice answer sheets. Student entries on the form give the student's social security number, the test identifier, and the answers to the questions in set fields across the paper. When submitted the answer sheet is processed through an IBM 1230 Optical Mark Reader which reproduces the student responses on a machine readable punched card.

The punched cards representing the tests are scored on the IBM J60/65 at the I.S.U. Computation Center. Student records kept by section and social security numbers are output in three formats. A magnetic tape is used to record the output of the current examination and later serves to be the starting input in the next exam. Three printed records are created. One is a cumulative record and is used by the instructor (Fig. 2). Another is a record of the results of the current examination plus the highest grade previously earned on a phase if there is no current score. Symbols appropriately designate the source of a grade (Fig. 3). The third printed record is an exception listing. Anytime an examination is scored a certain percentage of students fail to or wrongly encode their social security number or key identifier. Students in this category are not scored but are placed on the exception listing which prints a character replicate of the punched card. When the record is identified it can be immediately hand graded and update commands given to the master student file by a maintainer program.

The maintainer program is set up to add, delete, or partially modify records in the student file. Input is through mark sense cards that can easily be encoded during student consultation and submitted at a later date.

2. Test Generator Process. Programs have also been developed to use the J60/65 in compilations of the exams. The previously prepared question pool was entered on magnetic tape. Questions were grouped by phase and by 10 topical areas within a phase. Each taped question had associated with it: (a) a unique identifier, (b) the question and responses, (c) the correct answer indicator, (d) a difficulty estimate. The unique identifier allows statistical information on a question to be stored for later evaluation.

Tests were constructed by generating random numbers which were used to choose a question in a topical area. Topical areas had one or two questions chosen from them to construct a phase test of 15 questions covering 10 topics. A key was simultaneously generated to be used in the scoring of the test. Each generated test had a cumulative difficulty rating derived from the sum of the difficulty of the questions. Instructors used this to screen overly difficult or easy tests. If a test was approved it was sent to the copy center where the required number of copies was made.

A flow diagram in Fig. 1B and 4 summarize these points.

With the subject material and grades modularized it is now possible to internally monitor the effectiveness of the teaching/ learning situation. Average scores earned on a phase can be compared thus creating a feedback loop to identify problem areas in student performance. Appropriate corrections than can be made in the objectives, testing, or instruction to insure subject material competence.

Advantages of PAS

The PAS has provided solutions to problems faced in teaching large numbers of students. The main advantages include the following:

1. students may progress at their own rate of learning through the course. Well prepared and capable students can pass phases early in the course, perhaps at the first meeting, whereas slow-learning, distracted, or ill prepared students can take all term plus another term if necessary. Thus, learning the material as well as in demonstrating how much one knows on an exam is self-paced.
2. Since the subject material and testing is modularized a student knows where his deficiencies in biology lie and can efficiently use his time on these areas.
3. The grade earned represents the mastery of an amount of biological information. A student is required to have a broad understanding of biology as well as depth in 7 out of 8 areas (Fig. 5, 6). This degree of knowledge is not necessarily assured under the traditional method of teaching.
4. because of the retesting provision, competition for grades among classmates is eliminated. Rather a student competes only with himself against standards set by the

instructor and outlined in the objectives. Grades reward student enterprise rather than intelligence.

5. The responsibility for learning rests on the student. The instructor is no longer in the central role, but rather is one of several sources of information.
6. Record keeping is facilitated, and instructors teaching great numbers of students are freed from this chore.
7. Time spent on exam compilation is minimized.
8. Computer listing allows for ready and easy accessibility of data for studies of basic teaching/learning problems and evaluation of subject areas within the course.
9. PAS is readily adaptable to courses other than biology. Plans are underway to adapt it to other large enrollment lecture section courses at Iowa State University such as those in the Departments of Earth Science and Library Science.

OPERATIONAL ASPECTS OF COMPUTER WRITTEN AND SCORED
FIRST YEAR COLLEGE ACCOUNTING PROGRESS EXAMINATIONS

MICHAEL BALDIGO

INDIANA UNIVERSITY
BLOOMINGTON, INDIANA 47401
TELEPHONE: (812) 339-1069

SUMMARY

The following observations are collected from the experiences of the author in using computer written and scored objective progress examinations for first year college accounting courses in 1970-71. These are not just classroom observations nor computational aspects of the data processing; the discussion centers on the administrative tasks of distribution, control, and student-system feedback in handling nearly 800 weekly exams for three concurrent accounting courses, each divided into numerous discussion sections. No firm conclusions arise from these observations, but a number of insights and guideposts are suggested as to the planning one would do in tapping the computer's vast information processing capabilities to meet more effectively one's own educational requirements.

Introduction

Progress examinations present a painful set of sacrifices in teaching effectively, particularly for required business program courses such as accounting A201-202 and for popular electives like accounting A200. Progress examinations, or a reasonable substitute where possible, are highly desirable in maintaining a student's interest, motivation, and sense of commitment and cumulative progress. Yet they are an increasingly high-cost, labor-intensive effort, which competes directly with time for instruction and preparation.

On the horns of this dilemma, the teacher can re-stretch his resources and attempt to write, grade, record, distribute, and discuss such exams by himself or with relatively little incremental assistance, watching the burden grow exponentially as enrollment redoubles; or he or she can cut back and watch the quality of the teaching-learning process fall off in giving, by necessity, fewer and fewer progress/incentive maintaining exams.

Avoiding these drastic alternatives is the foundation of the work behind this paper. Admittedly, the computer isn't perfect; it is no substitute for good teaching by anyone's estimation. But it may provide a very effective clerical substitute for the teacher, augmenting his effective time and skills in this area.

Electronic computational capabilities allow us to get more out of teaching resources, which may already be stretched beyond what is desirable or maintainable in the long run. If we are particularly watchful of the pitfalls, responsive to the challenges, and aware that the computer cannot do the whole job single handedly, much can be done to avoid a drastic stretching of resources and/or lowering of educational quality.

In a deceptively simple statement, the goal we are working toward is to use computer information processing technology to improve the cost-effectiveness of teaching basic accounting, administering weekly progress examinations by a computer system. From non-classroom and non-data processing experiences, here are operational aspects of how carrying out such a computer/teaching system worked.

The Basic Computer System

For each semester accounting course, large groups of questions for each week's materials were collected. Generally graduate students in accounting were used to write questions utilizing previous testing materials. These together with their approved answers, were placed in a computer data base, then selected at random to create ten question examinations which were printed out by the computer. These exams were numbered sequentially for each course and each was given a grading code to be used as an input to the computerized grading and scoring program.

Each Monday night from 4 to 10 PM, students received a computer written exam paper and took the exam by noting his course, name, student ID number, answers, and grading code onto machine readable answer sheets. Keeping their questions (and having been instructed to indicate their final answers on their questions), students turned in their answer sheets, which were electronically scanned and graded, the results recorded, printed up, and posted for verification by student ID number.

In order to provide prompt, adequate knowledge of results, the ten approved answers for each exam were printed up, and these were handed out by matching with the exam number as students presented identification and turned in their computer answer sheets. Thus it was not necessary for students to surrender their questions and wonder about correct answers, nor did we have to return their self-graded score with the computer printed results that were posted.

Student Interface

The non-classroom elements of student interface were the distributing of exams for each course as students entered the exam room, proctoring the exam room (copying was out of the question since each student had an individual form of the examination), checking ID cards, collecting and visually inspecting answer sheets for oversights, and handing out the correct answers to match each exam. Although students could take as much time as they wanted during the exam period, handing out the exams in sequence corresponded approximately to the sequence of answers needed on departure (with a few bottlenecks), and no major confusions resulted while this procedure was employed.

Also, as the system evolved, there was further student interface of major importance in regrading questions students wished to challenge, to Request for Credit (RC) forms. All phases, except RC evaluations, could be handled entirely out of class by non-teaching staff.

Problems Encountered

The first problems encountered were in spacing and scheduling the examination system which had served several hundred students within a fairly short interval of time. At first, almost nobody showed up during meal hours, and the last hour produced explosive queueing problems. Many of these most obvious problems were self-correcting: after waiting forty-five minutes, many students set out to arrive earlier or at less popular times; furthermore, service rates improved as students and staff became more familiar with the procedures.

Cheating within the exam room was not a serious problem and monitoring was rarely necessary. Occasionally partners would swap examinations, although this was blatantly obvious to students and monitors alike and was a high risk strategy tried by very few. Controls could have been instituted to insure that the same exam number and student number stayed together until turned in, but this was thought to be costly and of dubious benefit under usual circumstances.

Cheating by sitting for one form of the exam, turning in a defaced or blank answer sheet, then retaking the exam with this knowledge was thought to be a minor problem. As soon as checkers learned, from catching errors in copying or omissions, to examine answer sheets at a glance before accepting them, the opportunity for such "double-taking" by turning in an ungradable answer sheet the first time was extremely small.

Occasionally an A201 student would take an A202 exam by mistake. This did happen to a few foreign students who passed blithely through the system imitating the fellow in front; however, problems like this were rare and almost always corrected before the student left the examination area.

Post-examination queueing, milling about, and answer swapping proved to be the worst initial problem. Having received their "approved" answers, to be compared with their answers, students frequently sat right down in the crowded hallway to compare, exchange, visit with students entering, argue out details, await for friends to finish, lament, or celebrate. For many students to do this all at once, the burden on the tightly moving system was too much, encouraging extreme cases of prior "approved answer" exchanges and frequently preventing others from entering or leaving easily.

After six hours of collecting exams and approved answers from the hands of friends, several people developed their own data bank of questions and answers, as well as a good familiarity with the parameters of the random selection program selecting the questions. This problem also exacerbated the difficulty of everybody showing up at the last minute, having waited to collect the news on all the tough questions in advance!

One could revise the examination period, develop a question selection program which released only part of the questions into the selection pool for the exams to be given out earliest, or institute harsher controls at various levels. However, semester schedules had been established based on the 4-10 open examination period and the burden of changeover would have been quite undesirable.

Instead, the procedure was instituted to give out answers on the following day, from 12 to 4 PM in a single room. Large scale "data banking" during the exam was eliminated; however, the more enterprising students with a strong working knowledge of the material in the first place

could still look up answers and, hopefully, improve their odds almost as much as studying an equivalent amount of time would have done!

Passing out answers the following day led to its own facilities problems. Three times the reasonably anticipated amount of space was needed. There was considerable scattering once sheets were stacked in groups of ten for self-selection of answers, it being impossible to hand answers out individually under the circumstances. Student cooperation in restacking the scattered answers helped alot, but often an angry student getting a bad grade left his stack of ten and five others in shambles, undoing the care and efforts of many others.

Unfortunately, one would think students would remember their exam numbers or keep their exam questions with them, but many did not. Confusion and poor communications in such a large group were hard problems to overcome; a number of students didn't really take the ten point examinations very seriously; and other factors are examined below.

Distributing examination answers in one room for three courses saved amply on staff and on confusion between designated pick-up points; however, this often led to picking up the right numbered answers to the wrong course. Loud complaints about missing answers were resolved, as a rule, by showing the student the pile for the correct course. On average, only one student per week was unable to secure his answers during the distribution session after making judicious efforts to find it. While a number of causes for missing answers was evident (friends picking up friends' answers and so forth) I was aware of nobody who couldn't find his answers after a day or two.

It was discouraging that students continued to forget numbers, look in wrong course stacks, and take somebody else's answers throughout the year. Although the frequency of such cases dropped by about 50% between the first and second answer distributions, there was little improvement beyond that stage. The principal cause was that not students, but students' friends, classmates, wives, and so forth were picking up the answers for them. The student might know precisely where the appropriate answers were stacked, his course number, exam number, and so forth (the placement of stacks was kept identical from week to week), but his girl friend stopping by on her lunch hour hardly knew her way to the business building! Other losses were due to carelessness, perversity, and semi-sabotage, though all were quite minor.

We found that A200 had few enough sections that the answers could be sorted out by the discussion section instructors themselves and given out at regular class meetings. This took a major burden off of the confusion, as we were left with only two courses for which answers were being self-selected. Unfortunately, students and their representatives, in dwindling but persistent numbers, still showed up to get A200 answers five weeks after the changeover.

The situations in A200 can teach us much on several counts. In similar systems, one would be wise to number A201 exams from 1-500, A202 from 501-1000, A200 from 1001-1500, since students remembered only the exam number not the course when picking up the results for others. Even when getting their own answers, students frequently didn't notice the conspicuously placed course number at the top of each page until they got home! Mutually exclusive numbering sequences would prevent this confusion.

Furthermore, if one could carry out the large administrative and educational process involved, differentiating discussion sections in handing out students' examinations would solve many problems. For example, allowing Section #1 A202 students exams numbered 500-549 and Section #2 A202 exams 550-599 by (a) giving the exams at different times or places for different sections, (b) checking section numbers in handing out individual exams, or (c) having students select their exam papers from boxes labeled for the appropriate section would do the task.

While adding to the costs, administrative burdens, and possible confusions, this system of linking exam numbers to discussion sections would eliminate altogether the need for a mass distribution room for securing one's answers. Answers could be given to discussion section instructors and distributed at the next discussion meeting. Depending on whether or not the answers were to be discussed, the individual discussion sections could work out ways to distribute answers with minimal disruption (i.e., self-selection upon entering or leaving class).

Problems With Questions

Examination questions produced considerable trouble on many accounts. First of all, they were not consistent as to directions because they were ordered at random. Using answer sheets that allowed for responses from A-Z or from 1-26, on question #1 a student might be asked to mark "A" for true and "B" for false; on question #2 the same student would be told to write "T" for true and "F" for false. Although there are pros and cons regarding severe penalties for not following directions explicitly, equity required that students be given credit when confusions of this nature were easy to make.

241

Second, the best written objective question can often be read in two ways. The "one and only one" correct answer, perfectly unambiguous to a person specialized in accounting at an advanced level has a perfectly "correct" meaning if looked at from another point of view. At times, when read by a student specializing in mathematical logic, a question could be technically false or ambiguous, though to any accountant it was true. For example, "Debits always equal credits in an accounting system"...but do they in the case of an undetected posting error? In theory the two must be equal; the answer is true. According to practice and strict logic, there are occasional exceptions; the answer is false.

Third, there was every incentive to raise a complaint against a debatably ambiguous question, as there was no penalty for an extremely weak Request for Credit (RC) slip. Often the stack was four inches thick at the end of a handing out session. A long, hard task of regrading was required, judging each complaint or counterexample on its own merits.

Such regrading was fundamentally unfair because one can hardly give points to all persons answering the debits equal credits example question incorrectly, some answering "B" instead of "F" which the computer accepted, some having brought up the logical technicality justifying "F", and many missing it due to not knowing the material. On the other hand, one can hardly throw out the question and regrade all examinations which (somewhere) have that question, using a nine point revised percentage scale. This is especially true when the exception to the approved answer is based on a special situation or well articulated technicality...which is the only way the different answer could be justified! In recounting points, only those submitting complaints could be given credit for alternative answers, which gave a major incentive to submit RC slips and exacerbated the problems in handling their large volume.

Fourth, many questions, being made by graduate students in accounting and/or with substantial experience (indoctrination?) in accounting, tended to concentrate on fine distinctions, which a basic accounting student might not grasp until he finished the year's work. Further, many questions asked almost the same thing in a slightly different way (perhaps an important one but nevertheless a confusing one). Complete consistency was impossible to attain with many question writers, and overlapping of similar points of theory led, at random, to ample confusion and distraction.

Especially when chapters assigned for the week covered a limited amount of material (for example introductory chapters) one quickly reached diminishing returns on writing different questions; yet the press of needing a large question pool to make the computer selection system work invited fine distinctions rather than a few, well-worded, well-groomed objective questions.

Fifth, questions had a painful tendency to hinge upon a single word like "always," "never," "every," or "only" to make them unambiguously one of the alternative answers. This is a fact of life with objective questions; however, it added to frustrations and complaints as students came to realize and resent that an astute guesser could answer difficult questions without any real knowledge of the material.

Sixth, students soon came to realize that ten questions selected at random from a large pool of hard and easy questions do not dependably measure one's performance on any single test. The questions could easily all focus on one topic, or they could scatter over areas which were not central to the readings. A large measure of chance came into every exam, and the connection between preparation and good performance was strained by the "luck/chance" factor. This is one key reason why some students did not take the examinations nor their results terribly seriously. More questions per exam (this year there are fifteen per exam), more consistency in the questions, and better communications that on average students' preparation will be well measured during the term are ways in which these problems can be remedied.

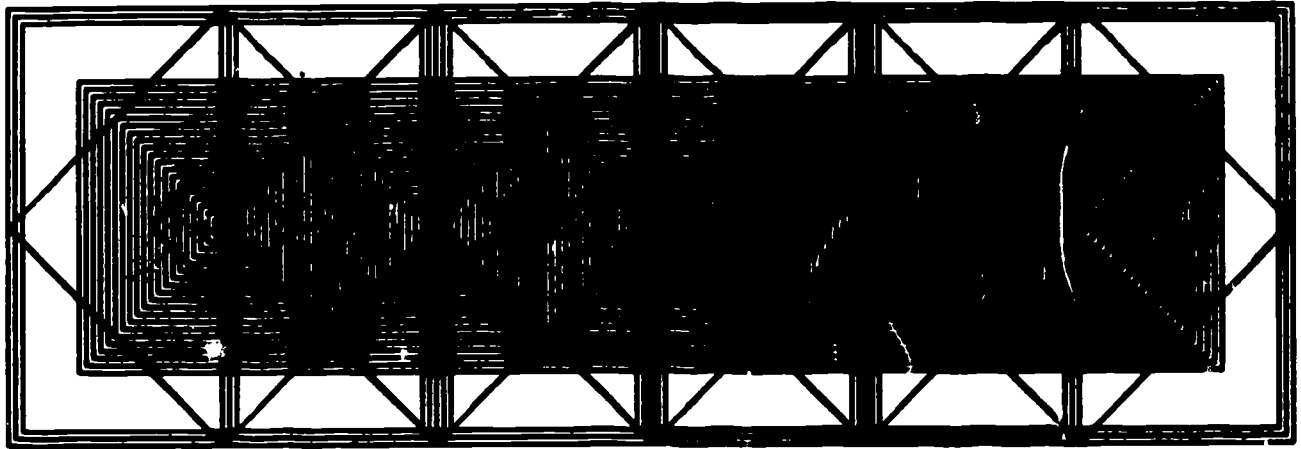
Conclusions

The computer teaching system achieved its major objectives, putting the computer to work in what was formerly a high-cost, labor-intensive process of grading which was rapidly pricing itself out of being a workable teaching tool. It is dubious that this amount of grading, which was not itself faultless, could have been duplicated with the same regularity and reliability using traditional teaching methods. While emphatically not a sure-fire panacea, please remember this qualification, computer written and scored progress/incentive examinations can be quite effective.

In working with similar systems for educational needs elsewhere, one will find computer systems like this one present a number of pitfalls which deserve planning and prompt remedial attention. Many parts of our system proved to be far more labor-intensive than reasonably anticipated; for example, evaluating the large number of answers challenged for good and bad cause.

Many inequities, alienations, and difficulties in communications tended to arise, and these were intensified by the complexities of the system. Problems such as excessive EC slips tended to go unchecked, and a solution such as reserving the right to penalize extremely weak EC slips to balance the incentives to submit them requires careful planning, communication, and consistency in enforcement to be effective.

Given the outcry for more effective uses of teaching resources, in assisting the teaching process with high powered clerical services of computer technology, we find a means of improving the cost-effectiveness of education. Admittedly less than perfect, such a computer system is a better alternative than either stretching teaching resources by unworkably labor-intensive examination procedures or reducing the commitment to frequent progress/incentive maintaining examinations.



Dorothy Holmes

DIMINISHING SQUARES by Dorothy Holmes

Problem: Diminishing polygon forms with serial imagery

ERIC 244₂₃₄

THE USE OF A COMPUTER FOR MOTIVATING STUDENT PROJECTS
IN UNDERGRADUATE COURSES ON NETWORK THEORY

Roland Priemer and Tadao Murata
University of Illinois - Chicago Circle
Chicago, Illinois 60680
Telephone: (312) 996-5491

ABSTRACT

This paper describes the use and content of a library of computer subroutines made available to undergraduate students in courses on network analysis and network synthesis. The primary concern in constructing this library was that it be organized to be versatile enough to allow more originality and initiative in student projects than existing computer programs for network analysis and synthesis.

Introduction

This paper is concerned with a library of computer subroutines designed to motivate student projects using a digital computer. Involved are students in a sequence of two senior level courses on network analysis and network synthesis at the University of Illinois, Chicago Circle. These courses naturally allow a meaningful application of the digital computer. The first course presents the topological approach to network analysis. The second course is concerned with various methods for designing networks from given network functions or desired network performance. The prerequisite for these courses is a background in introductory circuit and linear system theory. Also as part of the engineering core curriculum, all students have taken a course on Fortran programming.

Incorporating the use of a computer in these courses has allowed the students to consider more meaningful and practical problems. There are many computer programs available for both network analysis and synthesis. Such programs are widely used in practice, and have become as commonplace as the oscilloscope or the slide rule. However, most of these programs are not flexible enough to be pedagogically useful in undergraduate courses on network theory. This statement is based on a number of observations (which are discussed in the sequel) made by the authors and their students.

In addition to the use of the computer, the authors also believe that the student greatly benefits from working independently on long term projects. By allowing for a more active participation in the learning process, a student's involvement in an individual or small group project helps to develop the initiative and self-reliance that he will be called upon to utilize when he becomes a practicing engineer.

Observations on Motivating Projects

Students at Chicago Circle have been using the computer in the above mentioned courses during the past several years. Indeed, almost all students have expressed a great desire to use the computer.* This use has been in the form of students writing their own programs or using large canned programs such as ECAP.

Students have expressed dissatisfaction with both of these approaches. Although students have been able to write their own programs, this approach has received the most serious criticism. This is due to the considerable amount of time required for writing and debugging programs. The dissatisfaction with canned programs arises primarily because they are not useful for understanding the theory on which these programs are based. Most students feel that neither using canned programs nor writing their own in the time allowed contributes significantly to understanding the techniques presented in these courses.*

The authors agree with a statement by Chua[3] that the use of a canned program "tends to create a strong student desire to depend heavily on canned automatic programs to solve even the simplest of problems." It was also found that canned programs do not motivate the student to seek an understanding of the theory or the operations occurring within the computer, even though these programs are used for dealing with practical problems. This reinforces the findings of Dertouzos[5].

* Student opinions are based on the results of questionnaires given to 120 students during the last two years.

With the above considerations, the authors constructed a library of computer subroutines based on the following guidelines.

1. The library should allow for a gradual increase in the use of the computer as the courses develop and material is presented.
2. It should be possible to satisfy a student's desire to participate in the programming effort.
3. The interested and capable student should be able to develop his own specialized programs by using the library as a basis or to augment the library.
4. It should be necessary for the student to use what he is learning in the course in order to use the library of computer subroutines.

This library has been used during the past year in the previously described courses.

Network Analysis Course Subroutine Library

Writing a program for the AC analysis of RLC networks is mostly an organizational problem rather than a numerical one, and requires a considerable effort of the student. The subroutines in the network analysis library represent the steps involved in applying the basic concepts to solving problems. The subdivision of tasks allocated to each subroutine has been done in such a way that the student cannot link together a set of subroutines to form an entire program unless he understands the basic theory involved. With these routines, a student's analysis program consists entirely of a set of "CALL" statements, which, based on an understanding of the concepts, he must properly sequence. In the appendix is a list and explanation of these routines.

For example, a student's loop current analysis program might be the following program.

```
COMPLEX VOLTS(30), AMPS(30)
DIMENSION X(100), Y(100)
CALL REDATA
CALL INCID
CALL RINCID
CALL TREE
CALL TIESET
DO 1 I = 1,15
CALL BRACHE
CALL ZLOOP
CALL ELOOP
CALL ILOOP (AMPS, F)
CALL BCURR (AMPS, F)
CALL IBTOVB (VOLTS, F)
X(I) = F
1 Y(I) = REAL (VOLTS(3))
CALL PLOT(15,X,Y,'HERTZbbb', 'VOLTSbbb', 'bbbbbbbbb
1 FILTEROUTPUTbVOLTAGEbVSbFREQ.bbbbbbbb', 'LOGb')
END
```

In this main program, the student wanted a plot of the magnitude of the voltage across branch 3 as it varied with the 15 frequencies that were read in. In addition to this output, most subroutines print out the results obtained within these routines. For example, ZLOOP prints out the loop impedance matrix as a function of s .

Since all information is transferred internally via common storage, the student can access any of the variables in these routines by the addition of a few "COMMON" statements to his main program. This will allow considerable variations in the use of these subroutines. For example, circuit information can be changed for a repeated analysis, any Fortran statements can be added to the main program. This will allow considerable variations in the use of these subroutines. For example, circuit information can be changed for a repeated analysis, any Fortran statements can be added to the main program, or the student can develop his own subroutines for other information. Also, any conceptually logical subset of these routines can be used to conform with progress in the course. By the end of the network analysis course, students have written nodal, loop, or node-pair analysis programs and are able to compare the capabilities and limitations of these different methods with respect to numerical accuracy and computation time.

Network Synthesis Course Subroutine Library

Reviewing past experience in guiding student projects in a network synthesis course, it becomes evident that the originality of a student design project and the extent to which projects are interesting and practically useful are greatly limited by numerical as well as programming difficulties. Also, the time required for programming tasks which are actually incidental to the insights sought in a project, restricts the types of projects students can work on. These considerations indicate a need for programs in some canned form.

In opposition to the above conclusion stands the difficulty of determining what operations these programs should perform. This problem arises because in a network synthesis course the material that is presented consists of many specialized techniques, which are applicable only to variously restricted problems.

The subroutines that are therefore in the synthesis course library can all be regarded as utility programs, and thereby intentionally do not eliminate all the programming effort that might be required of the student. Rather, some of these routines eliminate tedious numerical and algebraic work, while others are the implementation of several numerical methods which may be essential to student design projects. The subroutines in this library perform such operations as:

1. Determine system functions of lumped, linear networks in symbolic form.
2. Obtain a partial fraction expansion of a rational function of s .
3. Calculate and plot the inverse Laplace transform of a rational function of s .
4. Obtain a frequency response, Bode plot, and Nyquist plot of a rational function of s .
5. Polynomial manipulation (addition, subtraction, multiplication, and division).
6. Evaluation of a polynomial from its zeros.
7. Function minimization.
8. Determine the roots of a polynomial.

In addition, each student who has taken the analysis course has written his own analysis program allowing for synthesis by repeated analysis.

Implementation of the Library

At the start of each of the courses previously described, each student receives a user manual. The manuals explain the purpose and limitations of each subroutine and how the library can be used. The library** is stored in on-line disk space.

For the first few weeks in the analysis course students are asked to call the subroutine SAMPLE. This routine is an example of what might be a student's main program, and it enables the student to become familiar with the computer center facilities. For the remaining part of the quarter, each student is assigned a project. This involves the writing of analysis programs using the library. These programs are then used, for example, to study the distortion due to loss factors in reactive elements or to determine the power delivered to a network.

The subroutine library for the synthesis course does require the student to do some programming. This adds a challenging factor to the projects assigned in the course. However, this library attempts to eliminate the programming effort required for necessary operations which are incidental to the insights sought in a project. Most projects involve a frequency or time domain synthesis technique. However, some students have developed their own projects, involving, for example, writing a program for obtaining system function in symbolic form using topological formulas or doing a sensitivity study of some active networks.

** A listing of the library of subroutines is available upon request.

Conclusion

This library of subroutines was intended to encourage the student to discover for himself how the computer can be used to implement the theories presented in the previously described courses. The authors therefore felt that it would be constructive to get some student reactions to the library. According to the questionnaire referred to earlier in the paper most students agreed that the library provided a useful tool for integrating the techniques presented in the courses and the use of the computer.

The authors believe that the library is helpful in three significant ways. Firstly, through its use, the student acquires a better understanding of the capabilities and limitations of the computer than through the use of canned programs. Secondly, the library described here requires the student to apply algorithms to problems in a systematic manner. Finally, utilization of the library allows the courses to be more independent project oriented such that projects are a vehicle for implementing the theory and developing the student's initiative.

ACKNOWLEDGEMENT

The authors wish to express their appreciation for the suggestions and comments offered by their students, and for the contribution to testing and debugging the programs made by A. C. Petersen.

REFERENCES

1. Cosine Committee, "Some specifications for a computer-oriented first course in electrical engineering," Commission on Engineering Education, Washington, D. C., September 1968.
2. N. Balabanian and T. A. Bickard, Electrical Network Theory. New York: John Wiley and Sons, Inc., 1969.
3. L. O. Chua, "A computer-oriented sophomore course on nonlinear circuit analysis," IEEE Trans. on Education, vol. E-12, no. 3, September, 1969.
4. J. L. Melsa, Computer Programs For Computational Assistance In the Study of Linear Control Theory. New York: McGraw-Hill, Inc., 1970.
5. M. L. Dertouzos, "Elements, systems, and computation: a five year experiment in combining networks, digital systems, and numerical techniques in the first course," IEEE Trans. on Education, vol. E-14, no. 4, Nov., 1971.
6. M. E. Van Valkenburg, Introduction To Modern Network Synthesis. New York: John Wiley and Sons, Inc., 1964.
7. P. M. Lin and G. E. Alderson, "SNAP-A computer program for generating symbolic network functions," TR-EE70-16, Purdue University, Lafayette, Indiana, August, 1970.

APPENDIX

The following is a list and brief explanation of the subroutines in the analysis course library.

- Subroutine SAMPLE - This is an example of an analysis program, which uses the library.
- Subroutine REDATA - This routine reads in the problem description, allocates information in storage, and writes problem and output description.

The following routines are used for the topological analysis:

- Subroutine INCID - Obtains the complete incidence matrix of a graph.
- Subroutine RINCID - Obtains the reduced incidence matrix of a graph.
- Subroutine TREE - Finds a tree of a graph.
- Subroutine TIESET - Determines the tieset set matrix.
- Subroutine CUTSET - Determines the cutset matrix.
- Subroutine ZLOOP - Obtains the loop impedance matrix $Z(S)$.

Subroutine YNODE - Obtains the node-pair admittance matrix $Y(S)$.
Subroutine YDATUM - Obtains the nodal admittance matrix $Y(S)$.

The following routines are used for the frequency analysis.

Subroutine BRACHZ - Determines the branch impedance matrix.
Subroutine BRACHY - Determines the branch admittance matrix.
Subroutine ELOOP - Obtains the loop voltage source matrix.
Subroutine JNODE - Obtains the node-pair current source matrix.
Subroutine JDATUM - Obtains the nodal current source matrix.
Subroutine ILOOP(AMPS, F) - Calculates the loop currents.
Subroutine VNODE(VOLTS, F) - Calculates the node-pair voltages.
Subroutine VDATUM(VOLTS, F) - Calculates the nodal voltages.
Subroutine BCURR(AMPS, F) - Obtains branch currents from loop currents.
Subroutine BVOLT(VOLTS, F) - Obtains branch voltages from node-pair voltages.
Subroutine VBTOIB(AMPS, F) - Determines branch currents from branch voltages.
Subroutine IBTOVB(VOLTS, F) - Determines branch voltages from branch currents.

With the above routines, a frequency analysis can be performed using a loop, nodal, or node-pair method. The following routines are support subroutines for the above routines.

Subroutine POLAR(A, N) - Converts complex numbers from rectangular to polar form.
Subroutine LINEAR(A, N) - Converts complex numbers from polar to rectangular form.
Subroutine PLOT(NPTS, X, Y, XUNIT, YUNIT, NAME, KEY) - Obtains a plot of the variables stored in Y versus those stored in X.

**A COURSE IN COMPUTER SIMULATION AND ANALYSIS
FOR SCIENTISTS AND ENGINEERS**

Bruce W. Peterson, George W. Questis and Arthur Houghton
The University of New Mexico
Albuquerque, New Mexico 87106
Telephone: (505) 277-4549

Introduction

In this paper, a course currently being taught in the College of Engineering at The University of New Mexico is described and discussed. The course is a one-semester survey of several of the large-scale analysis programs currently in use throughout the country. The course is oriented to a general audience of students from several disciplines such as Chemical Engineering, Mechanical Engineering, and Nuclear Engineering. It is taught by a team of three professors of various disciplines in separate contiguous portions of the semester.

The particular analysis programs studied are CSNP, Continuous System Modelling Program; ECAP, Electronic Circuit Analysis Program; CINDA, Chrysler Improved Numerical Differential Analyzer; CHESS, Chemical Engineering Simulation System; and CONFAC II, a computer program for the determination of radiant interchange geometric form factors. All students are briefly introduced to each of these programs, and each student is given an opportunity for some advanced work with one of the programs by means of a term project.

Objectives of the course

The primary directional objectives of the UNM College of Engineering in teaching "computers" to its students can be summarized as follows:

1. To give the student a comprehensive working knowledge of FORTRAN.
2. To teach the student the methods and use of large "package" programs.
3. To teach the student to take a program currently in use elsewhere and make it work on the computer system he is using.

The course is intended to help fill part of the deficiencies in number two and, in some cases, number three. The need to introduce students to package programs is emphasized by the fact that, among 20 engineering professors attending a recent engineering design seminar at Stanford, only one, George Questis, knew how to use any of the Continuous System Simulation Languages, even though they are simple to use and greatly enhance a person's problem solving capability. The course is also intended to reach both students who are applications oriented and students who are computer science oriented and interested more in methods than in results.

There are a large number of analysis programs in use. Since it is infeasible to expose the student to all of them, some cross section of the available programs must be chosen. It is desirable to let the student compare the different programs with regard to the following:

1. Program types and potential use.
2. Methods of problem setup and data input.
3. Techniques used in analysis.

A good way to choose programs which have variety in the above three points is to choose both general purpose programs, such as CSNP, and special purpose programs, such as CHESS, as well as to choose programs from various disciplines. Keeping the above in mind, the final decision to teach a particular program can be based on the following criteria of a good program. The primary prerequisite is that the program be available and convenient to the student so that he can get several turn-arounds per week. This is most important to the learning process. Secondly, the program should have a single input language which describes the problem structure, states element or parameter types and values, specifies the excitations, and indicates the analysis options desired. Third, it should be able to handle nonlinearities. This is usually done by either a piecewise linear approach or by a mathematical subroutine approach. Fourth, a variety of output options should be available, such as transient response, steady-state response, plots, etc. Fifth, the program should have the capability for automatic parameter modification with a solution being found for each parameter value. Finally, the program must have self-contained error-checks and diagnostics.

Few if any programs will satisfy all of the above criteria. For instance, a special purpose program may not handle a wide class of problems.

Based on the above criteria and the experience of the teaching professors, the programs named in the introduction were selected. The following section is a discussion of those programs.

Characteristics of the programs chosen for the course

The S/360 Continuous Modeling Program (S/360 CSMP) is only one of a class of problem-oriented programs, or languages, designed to facilitate the simulation of continuous processes on large-scale digital computers (see Table 1). The class to which S/360 CSMP belongs is referred to as continuous system simulation languages (CSSL) following specifications set up in 1967 by the software committee of Simulation Councils, Inc. Prior to that time, major computer vendors each offered their own version of such a program to run exclusively on their machine. The CSSL specifications have prevented divergence by selecting best qualities of several available programs toward which vendors might concentrate in future development. A language called CSSL II has since been developed which closely follows the specifications and runs on several different vendors' large-scale machines.

TABLE 1. TYPICAL CSSL PROGRAMS

<u>Name</u>	<u>Vendor</u>	<u>Machine</u>
S/360 CSMP	Int'l Business Machines	IBM S/360
MINIC	Int'l Business Machines	7090, 7094
	Control Data	CDC 6400, 6600
	Rand	Univac 1106, 1107
	Xerox Data Systems	SDS Sigma Series
CSSL III	Programming Science	Univac 1108, IBM S/360

CSSL programs are general-purpose tools primarily used in science and engineering fields for the analysis of physical systems which may be described by ordinary differential equations. These programs are basically a set of functional blocks combined with a centralized numerical integration scheme and a flexible language for communication. They feature ease of programming with input statements capable of being prepared from either block diagrams or differential equation notation. The user may construct his own functional blocks, or incorporate FORTRAN, to tailor the program for his own special purpose.

Consequently, typical applications have ranged from simulation of the human cardiovascular system by physiologists to global ocean current simulation by meteorologists. The universal capabilities of the program are in evidence in proceedings of the highly successful Summer Simulation Conference (Denver, 1970; Boston, 1971).

Because of these features, CSMP has been received by all students with overwhelming acceptance. The student with minimal computer knowledge quickly overcomes inhibitions as he finds he can readily model a system and achieve success with the program. The more experienced FORTRAN programmer marvels at the convenience CSMP affords, such as the provision for statement sorting to be done by the program, and not required by the user.

The engineering student quickly adapts CSMP to many examples from his own field, using it in his coursework or research. It has proven a boon to the instructor because the student is more involved with physical system modeling and less with programming difficulties. Further, the logic of users' simulation model can be easily scrutinized, which has proven difficult with languages such as FORTRAN.

Teaching CSMP to a mixed group of students along with other types of applications programs has demonstrated the need for user-convenience in all such programs. This has been appreciated by the engineer-user as well as the computer science major who may some day assume responsibility for developing or maintaining similar systems.

The contrast of the general-purpose CSMP with special-purpose applications programs has emphasized the advantages and power of the former. Limitations of the special-purpose programs are often frustrating, and the students find modification a formidable task. On the other hand, if CSMP does not have the capabilities desired, it can readily be combined with other applications programs. An example is combination of a CSMP model of an ethylene production plant with a scheme to determine optimal design using the IBM Linear Programming Package (MPS) [ref. 2]. Another example is the development of the CHESS Program allowing it to generate CSMP input statements [ref. 3]. In this manner, the blocks in the CHESS program which represent a steady-state chemical process may be examined in detail using CSMP to define their unsteady-state characteristics.

Problems are assigned to demonstrate the facility of CSMP for simulating systems resulting in:

1. Non-linear differential equations with variable coefficients (see Fig. 1).
2. Sets of coupled non-linear differential equations.
3. Finite difference equations for the solution of distributed parameter problems.

For such problems, a numerical approach is usually dictated, and CSMP greatly shortens the programming effort.

In the chemical industry, the design of chemical processes requires knowledge of overall material and energy balances for a complete plant project. A number of "flowsheet simulator" programs have appeared in recent years to promote computer-aided design ventures, such as General Electric Apache and Monsanto's Flowtran.

In such systems, a network of process equipment modules is constructed in which physical or chemical transformations take place under the influence of variations in thermodynamic parameters. In the programs, subroutines exist for various types of process equipment. The necessary physical parameters are generated by appropriate correlations from a library of thermodynamic and transport properties.

The mathematical problem inherent in models of such process systems is the required solution of large sets of simultaneous, non-linear, algebraic equations.

The CHESS Chemical Engineering Simulation System is currently available from the University of Houston at a nominal fee. It is worthy of note that CHESS is not as powerful as other commercially available programs whose cost is indicative of their position in a competitive market.

A CHESS simulation requires input information on feed streams to the process, structure of the network, algorithms for each process module, and program control parameters. The program output contains (1) information on each interconnecting stream, such as temperature, pressure, heat content, fraction as vapor and chemical composition, and (2) calculated values of certain equipment design parameters, e.g. energy required for a pump.

The value of a program such as CHESS lies in the capability of generating information for a number of design cases. Others have combined optimization schemes and cost estimation packages with these programs. In this manner, a minimal path search is made for a set of optimal design parameters. The output then includes the cost information which is so vital to any engineering design project.

The electronic circuit analysis program, ECAP, is oriented toward electrical engineers, but is by no means limited to problems in EE. Many problems in heat transfer, thermal radiation, and mechanics, to name just a few, have direct analogs in electronic circuits. ECAP can be used to analyze these analogs. Since solution of problems by use of analogs is a widespread approach, this is an important concept to teach the student.

ECAP uses a piecewise linear approach to the solution of non-linear problems. This is somewhat different than most analysis programs and offers a dramatic comparison to the student when juxtaposed with the other programs taught.

The input language of ECAP, which allows for the topological description of networks, provides an important link with other circuit analysis programs and is an easy-to-use method for the student-user. Besides the input language, ECAP is divided into three convenient sections: DC analysis, AC analysis, and transient analysis. This allows the student to quickly grasp the basic programming necessary. The user-oriented student can then advance to more complex techniques whereas the computer science oriented student can concentrate on the techniques used by ECAP.

ECAP has an automatic parameter modification feature which allows for variation of any of the elements in the DC and AC portions and for variation of frequency in the AC portion. Modification is not allowed in the transient analysis portion.

There is also a self-contained error-check and diagnostic system in ECAP which is a great aid to the novice. Thus, it is seen that ECAP satisfies many of the criteria for a good analysis program as defined in the preceding section and is, therefore, an appropriate language to teach in this course.

The following is a sample problem which all students in the class are expected to solve using ECAP:

1.

CSMP Exam Problem - Open Book & Notes - Time Allotted: 50 min.
 Objective: Formulate a CSMP Program, as complete as the given data allows, for the problem given below which describes a mathematical model for a rocket flight to be simulated on the digital computer.

Rocket Flight Model (Ref. James, Smith & Wolford. Digital Computer Methods. Note: Fortran solution given there using Milne Numerical Method.)

A small rocket having an initial weight of 3000 lbs., including 2400 lbs. of fuel, is fired vertically upward. The rocket burns fuel at a constant rate of 40 lbs./second, which provides a constant thrust of 7000 lbs. until the fuel is consumed, at which time the thrust is of course reduced to zero. Experimental data on this rocket are available for this analysis in the form of acceleration, velocity, and displacement versus time plots (Curve A, Curve V and Curve D). It is desired to use this data to check the output of the theoretical model.

Referring to the figure below, and assuming drag force proportional to the square of the velocity ($D=kv^2$) and constant drag coefficient ($K=0.008 \text{ lb-sec}^2/\text{ft}^2$), one might use Newton's second law of motion ($F=ma$) to obtain the following second-order non-linear differential equation with variable coefficients:

$$\ddot{y} = \frac{gT}{W} - \frac{K(\dot{y})^2}{W} - g \quad \text{with Initial Conditions (t=0)}$$

where $\ddot{y} = \frac{d^2y}{dt^2}$ and $\dot{y} = \frac{dy}{dt}$

T = thrust force, 7000 lbs. (for duration of fuel)

W = total weight of rocket and fuel at time, t, = (3000 - 40t) lbs.

K = drag coefficient = 0.008 lb-sec²/ft²

g = gravitational constant = 32.2 ft/sec/sec

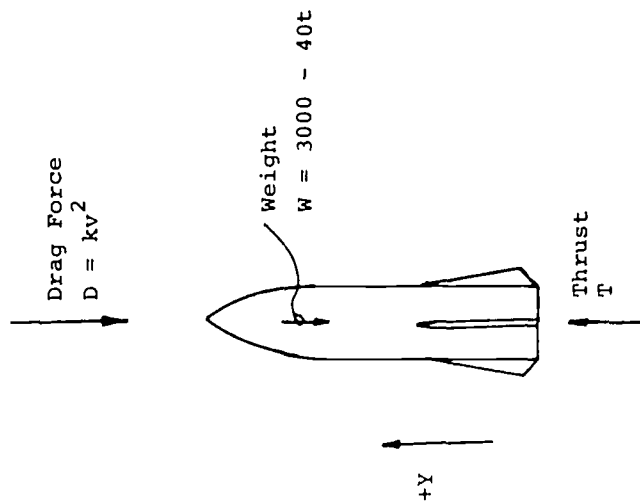
y = rocket displacement, ft., being positive upward from zero

\dot{y} = rocket velocity, ft/sec

\ddot{y} = rocket acceleration, ft/sec/sec

Programming Details:

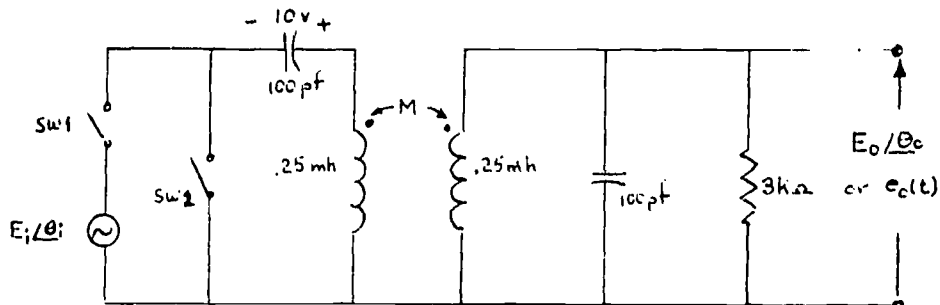
1. It is desired to run the problem for at least 200 seconds or until the rocket displacement returns to zero, whichever is less.
2. Integration interval is 1.0 seconds, with output every 2.0 seconds via Printer Plot option. Use MILNE Integration.
3. Output should give computed acceleration of rocket versus time (Printer Plot) with the experimental value printed on the same sheet, along with the error, i.e. difference between computed and experimental values.
4. Output for both velocity and displacement of the rocket should be given in the same manner as acceleration in (3) above.



Free-Body Diagram of Rocket

FIGURE 1. Nonlinear Differential Equations with Variable Coefficients.

- a. Determine the frequency response of the double-tuned circuit shown below for coupling coefficient values of $K=0.3$, 0.5 , and 0.8 ($M=K\sqrt{L_1L_2}$). Plot the gain, E_o/E_i , in decibels and the phase shift, $(\theta_o - \theta_i)$ in degrees. Assume switch 1 is closed and switch 2 is open.
- b. Assume both switches are open and C_1 has an initial voltage of 10 volts. Determine the transient response $e_o(t)$, for the 5 μsec period after switch 2 is closed. Use $K=0.3$.



One of the basic objectives of the course is to provide familiarization with computer techniques that are currently used in industry. Some of these problems involve various differencing methods, the use of automated and conveniently referenced information filing systems, and the interaction between complex sets of subroutines. In order to provide all of these features and to allow the student to become familiar with their utilization, the CINDA code has been chosen as a working tool. The fundamental mathematics is easily described and the student is led through one or two simple problems before being requested to perform a preselected elementary solution. After that elementary solution has been successfully performed, the student is allowed to select a somewhat more complex model.

A comparison of the different types of physical problems which might be encountered allows the student to see the total applicability of this type of differential equation solver. The student is not required to obtain a detailed understanding of all of the 350 subroutines which are routinely available, but simply must understand the basic analysis technique, the rudimentary flow diagram, and the general philosophy of interconnection of the elements. The student is assisted in developing the ability to read the instruction manual and finally is allowed to select a relatively large problem so that he can complete a solution which relates, in a reasonable fashion, to the types of things that employers might expect in the application of modern technology to the solution of real time-varying scalar field problems.

A more limited objective is chosen in the CONFAC analysis. This is a simple numerical method for determination of geometric radiance-interchange factors used in radiative heat transfer and illumination engineering. The program chosen is not particularly more appropriate than other generalized analysis systems, but the student is able to quickly solve elementary problems of a general nature and can readily perceive the broad geometric implications of this analysis. A simple geometric problem is solved as an example, and the student is encouraged to experiment with the utilization of the system. A relatively brief amount of time is allocated to this method since the generality is usually vested in the geometric representation of the physical problem.

Results

Responses of students taking the new course offering was good, judging from the amount of outside work which was accomplished. Most students attempted to solve problems using the programs (taught in the course) in conjunction with outside work. One example was a student taking a course in modeling biomedical systems. After running a CSMP program from the current literature which simulated the human temperature control system [ref. 1], he applied CSMP to simulate a muscle response model under discussion by his other class. The success of this application sparked sufficient interest to warrant an invitation to Dr. Quentin to speak to the biomedical class on CSMP.

Initial offerings uncover certain difficulties, and our course was no exception. In assignment of class periods among the three instructors, the semester was merely broken into thirds. Each instructor then taught consecutive periods until his material was covered. This proved not to be a convenience to the student who found it difficult to keep his program output in pace with class discussion. This was not student failure as much as computer turn-around difficulty. A related hindrance was that CINDA was not running at the time on The University of

New Mexico Computing Center IBM 360/67 computer. Consequently, student program decks were hand-carried to one of the local government laboratories where machine time was donated for running CINDA on a CDC 6600.

One of the benefits lost was that, by the time students had achieved a modest level of understanding of a program from running a few problems, the number of class periods devoted to that program was completed.

A student's capsule summary of the programs offered is given below:

CSNP	a quick way to computational relief and unsorted understanding
ECAP	the electronics designer's computerized edition of "How to Analyze"
CINDA	a forward-backward numerical erector set for hours of fun, profit and insanity
CONPAC	handy self-taught programmed course in coordinate geometry for rotational and translational freaks
CHES	less said the better

Future Plans

As discussed under results, the sequential assignment of class periods among instructors did not provide sufficient continuity for the student. To remedy this, it is planned, during the second offering, to assign one period per week (out of three) to each instructor. This means running three programs in parallel. Students have pointed out that it is feasible to keep three different problems active simultaneously, thereby improving computer turn-around.

The Engineering College has also been fortunate in obtaining an IBM 1130 computer to serve primarily as a remote batch terminal to the computing center IBM 360/67. This improves input/output conditions for the college considerably since the computing center is some distance away on the other side of campus. The 1130 computer also will serve as a stand-alone computer during certain periods of the day as well as times when the 360 is unavailable such as during maintenance periods.

A feature of the 1130 is the availability of an 1130/CSNP package. While this version of CSNP is not as extensive as the 360 version, it does provide interactive capability. Both versions of CSNP will be taught during the next offering of the course.

It was concluded that CONPAC and CHES were probably too specialized and had input languages which were too structured to meet the objectives of the course. In the next offering, these will be replaced with WASTRAN (NASA Stress Analysis Program) and SCEPTRE (a program for circuit and system analysis). There is no guarantee that this is the optimal selection. As previously stated, there are many possible programs which could meet the objectives. Some considered most worthy by the authors are ICES, GPSS, GASP, and DYWID.

While the course, as described, has filled a void in the curriculum, it is considered by the authors that the method used in teaching the course is most meritorious. The authors have visions of the course being offered at some future time as a simulation laboratory with the individual student deciding which programs he wishes to study and how many credit hours for which he wishes to register. This would allow maximum flexibility for students of any orientation.

REFERENCES

1. Winton, H. J., and Linebarger, R. W., Computer Simulation of Human Temperature Control, Simulation J., Nov. 1970.
2. Utsumi, T., Stone and Webster All-Purpose Simulator and Optimizer, and Its Applications, Proceedings of Conference on Applications of Continuous System Simulation Languages, San Francisco, July 1969.
3. Ingels, D. N., and Notard, R. L., A Process Dynamics Pre-Processor for CSNP, Proceedings of Summer Computer Simulation Conference, Denver, June, 1970.
4. IBM System/360 Continuous System Modeling Program User's Manual, No. GH20-0367-3.
5. Notard, R. L., and Lee, H. H., CHES Chemical Engineering Simulation System User's Guide, Third Edition, University of Houston, Sept. 1971.

6. Chrysler Corporation, Chrysler Improved Numerical Differencing Analyzer for Third Generation Computers, Space Division Technical Note TN-AP-67-287, New Orleans, Oct. 1967.
7. Houghton, A. V., and Edwards, J., CONFAC II, The IBM 360 Version of a General Computer Program for the Determination of Radiant Interchange Geometric Form Factors, Mech. Engr. Dept., Univ. of New Mexico, Albuquerque, Jan. 1971.
8. Jensen, R. W., and Lieberman, H., IBM Electronic Circuit Analysis Program, Prentice-Hall, 1969.

REALIZATION AND CLASSROOM APPLICATION OF A DISPLAY BASED SYSTEM FOR A SMALL DIGITAL COMPUTER

Donald C. Amoss and John M. Gowdy
Clemson University
Clemson, South Carolina 29631
Telephone: (803) 656-3379

Introduction

An introductory course in the computer sciences should include such topics as logical structure of digital systems, machine organization, information flow, data transfers, communication with external devices, and interrelationships between hardware and software. Typically, such a course must deal heavily with machine and assembly codes to the point where sufficient proficiency is achieved to allow demonstration of the other points mentioned. Several approaches could be taken, each with its inherent advantages and disadvantages. A hypothetical language could be used, designed optimally for the course being taught, but this eliminates any possibility of what is learned being directly applicable to any real world computer and, in addition, usually inspires little enthusiasm in the students. To use a large machine's assembly code would be very difficult because of the extensive instruction set and the problems that often develop when running poorly written machine code programs on a large system attempting to handle these jobs in the normal job stream. Therefore, it would seem that a real, small computer assembly language would be nearly ideal for such a course. However, there are still at least two alternatives. A simulator for the small machine could be run on a large machine with normal program submittal procedures as with higher level languages. There are certainly distinct advantages to this approach when dealing with large numbers of students; in fact, large enough numbers may preclude any other approach. However, eventually the students' approach to problem solving for this course would become identical to courses dealing with higher level languages; only the language would be different. There is a lot to be learned when forced to "run" your own computer that is not appreciated by those who submit a deck of cards and impatiently wait for the neatly folded answers. It is, perhaps, particularly important for future electrical engineers to have literally "hands on" experience with the collection of electronic circuits called a digital computer. This opportunity is enthusiastically embraced by most students.

If the decision is made that "hands on," real use of a small computer is an important feature to be included in the computer portion of an electrical engineer's education, then the question arises as to what can be done to optimize his experience without excessive cost or total investment. Any attempt to use punched paper tape as the only bulk storage medium would meet with immediate failure. On the PDP-8, one complete cycle of editing, assembly, and test execution with the required loading of the editor, assembler, and user program (the latter, several times per cycle) requires approximately one hour. (This is obviously dependent on user program length, but loading the editor and assembler requires over 30 minutes.) Some other form of bulk storage such as magnetic tape or disk is obviously necessary. Magnetic tape is less expensive and allows students to keep their own programs and operating system if they so desire. The system to be described uses magnetic tape bulk storage. Some comments should be made regarding existing systems for this computer using magnetic tape. Perhaps the most important criticism of existing systems is that they are not a system, but rather one program with a directory for calling all other programs including the assembler and editor. Since each program is handled independently, no internal intercommunication exists, and file names must be specified to every program called. Since the students will not require the greater flexibility possible with such an arrangement, there is no advantage for them. To the contrary, since the students will almost exclusively be working on one user program at a time and repetitively calling the editor, assembler, and loader, it would be less convenient and slower. The assembler should assume the program in the editor is the one to be assembled. When finished it should recall the editor with the same program already there either for modification and correction, or loading and execution.

The other major problem encountered on small computers is the usual output device—the teletype. Although the need for the teletype was not eliminated in the system to be described, a large portion of the intermediate use has been eliminated and replaced with a display scope. The only major time consuming task remaining is the listing of programs. Some thought is presently being made with regard to this problem and the possible addition of the line printer capability to eliminate this problem. Of course, the total investment for a complete installation would be significantly increased.

System Description

Based upon these primary objectives, SCPSYS (SCOPE SYStem) was written. The result was an interactive CRT/keyboard operating system capable of quick user response in performing editing,

assembling, and filing on a 4K, 12 bit word PDP-8 computer. The system's capabilities can be almost completely summarized by the command set. A brief description of these commands will follow a description of a few of the fundamental features of the system. All input is via the teletype keyboard or reader. Once the system is loaded, all operations can be performed via the keyboard with the exception of mounting different tapes, user program halts, tape hardware malfunctions, etc. That is, the system does not use the switch register or any other input during its normal operation; SCPSYS does not use the interrupt. The primary output device is an oscilloscope. (Our system has a 12" x 12" X-Y scope, but any oscilloscope of fast enough response time (10 MHz) and with Z-axis modulation should be adequate.) The teletype output echo can be suppressed (see commands listed later). The system has within it working areas for the current manuscript or source program and the current binary or object program. Whenever the system is called (or recalled) it displays the last several lines of the manuscript working area. (The actual number of lines displayed is adjustable by the operator.) Any portion of the manuscript working area can be displayed through proper use of the commands. All editing is performed with regard to what is being displayed. Nothing can be deleted from the working area that is not being displayed. The display output is much more rapid than the teletype oriented editors so students can "see" more of their program without excessive time consumption. This greatly increases the user's confidence in what he is doing to his program.

The minimum hardware configuration for using SCPSYS is a PDP-8 computer with:

1. 4096-word, 12-bit core memory
2. Extended Arithmetic Element (EAE)
3. TCO1 DEC-tape control
4. One TU55 magnetic tape transport
5. Two digital-to-analog converters and buffers (D/A)
6. CRT with appropriate amplifiers for D/A
7. ASR33 Teletype

It should be noted that SCPSYS operates more efficiently with two (or more) TU55 tape units, but only one is required. The estimated cost for a minimal system configuration is \$14,000. It may be advisable to acquire a system with a more flexible magnetic tape controller for an estimated \$21,000.

SCPSYS is stored on a standard DEC-tape and is loaded into the computer in the same manner as other DEC-tape oriented systems. The system occupies approximately 22% of the tape with the remainder available for user programs. If a second transport is available a file tape can be mounted on it which has over 99% of its space available for user programs. The index provides file oriented storage for manuscript (source) and binary (object) programs of the same or different names (eight character file identifiers.)

To perform functions within the operating system, a set of "executive commands" has been established. All other keyboard reader input is assumed to be new manuscript to be added to the existing working area manuscript. These commands are requested by starting a new line with a special character (>) which causes the rest of that line to be displayed (preceded by an arrow, →) at the bottom of the screen. A two character code is used to identify the desired action to be taken followed by any required parameters. A convenient set of default conditions exists which provide a means for rapidly assembling and executing programs during the debugging stages; no filing is necessary until desired.

To complete the discussion of the system features, each of the available commands will be briefly described. The commands can be divided into three categories; editing, filing, and assembly and execution.

Editing

The editor is the central program in SCPSYS in that all other operations are initiated from within the editor. It is capable of handling nearly 50,000 characters (more than sufficient for student use) with no provisions for handling larger programs in manuscript form. The manuscript is entered via the teletype after which it can be modified by adding or deleting characters, lines, or large segments with the aid of the following commands.

1. Line calls: A request to make a specified line the current display line so that some editing operation can be performed. For fine adjustment, a sequence of two characters is used to reposition the display backward or forward a line or a frame at a time. (A frame is equal to the number of lines being displayed; usually six.)
2. Add Manuscript: A request to add a manuscript previously filed by SCPSYS to the working area. The manuscript is added at the location in the working area at the time the command is given; beginning, middle, or end.

3. **Print Manuscript:** A request to list the contents of the working area or a specified manuscript file. Portions of manuscripts may also be listed by specifying desired range of line numbers.
4. **Erase Manuscript:** A request to erase a portion or all of the working manuscript area. Display can be anywhere within the working area when this command is given. A single line can be erased by requesting that line to be displayed and typing RUBOUT.
5. **Search for Character String:** A request to locate a specified character string (limited to eight characters) within the working manuscript area. Since tagged lines are usually preceded by the symbol "@" for indenting purposes (see discussion of assembling), the search can be limited to tagged variables or unrestricted to any use of the string.
6. **Character Editor:** A request to perform single character modifications within the last line being displayed. When the desired location within the lines is reached, erasing or adding characters is on a one character at a time basis. No editing takes place that is not visible on the display; i.e., only characters displayed can be erased and insertion of a new character is immediately displayed.
7. **Display Symbols:** A request to display the alphabetized list of the symbol table of the last program assembled. This aids in the debugging stages to confirm that a new symbol to be use is not currently in use.
8. **Set Display Lines:** Adjusts the lines per frame to the value indicated.
9. **Disable Typing Echo:** A request to eliminate echo of teletype input. Particularly advantageous when inputting programs from punched tape.
10. **Enable Typing Echo:** A request to provide echo of teletype input.
11. **Select Character Size:** A request to set size of characters in display. Of some value during demonstrations to provide larger, more readable characters.

Filing

1. **Save Manuscript:** A request to create a file and store the contents of the working manuscript area as a separate user file.
2. **Save Binary:** A request to create a file and store the contents of the working binary area (object programs) as a separate user file.
3. **Copy Manuscript:** A request to transfer a manuscript file from one tape to another.
4. **Copy Binary:** A request to transfer a binary file from one tape to another.
5. **Display Index:** A request to display index of programs file on indicated tape. Programs (either binary or manuscript) may be deleted at this time but two separate and deliberate acts are required.
6. **Print Index:** A request to print (hardcopy) index of programs filed on indicated tape. Some additional information is also typed with regard to filing space still available on tape.
7. **Create Index:** A request to initialize a new index on a tape that does not have one.

Assembling

Aside from a few features to be discussed below, the assembler is the same as PAL III Symbolic Assembler (8-3-5). A "text mode" has been added to provide a convenient means to store character strings for user program output. As mentioned above, the character "@" at the beginning of a line is not "seen" by the assembler but is used in all output, both display and hard copy, to provide indenting for easier reading of program listings.

1. **Convert:** A request to assemble specified manuscript (Passes 1 & 2 of PAL III assembler). Resulting binary (object) program is stored in the Binary Working area. An alphabetized symbol table is displayed upon completion.
2. **List:** A request to provide a complete assembly listing of specified manuscript (Passes 1 & 3 of PAL III assembler). Listing usually begins with the symbol table but it can be deleted.

3. **Add Binary:** A request to concatenate a filed binary program with the binary working area leaving the result in the working area. This provides an easy means for overlaying programs, or adding subroutines or data to user programs.
4. **Load Binary Program:** A request to load and execute a specified binary program. All of memory, except the bootstrap area, can be loaded via this command. Starting address can be specified at load time, defaulted to 200, or taken from index as the value specified when program was filed.
5. **Exit:** A request to store existing status on tape and put all loaders in memory ready for the next user.
6. **User Defined Commands:** A request to perform some user defined command. The command "7f" causes the system to load and execute four blocks of user written program which can branch on the contents of the accumulator which is set equal to the character after F in the command. This provides some flexibility in modifying the system and adding functions often used by specific installations.

In addition to the main system just described, there are several auxiliary programs which make use of the display based features to provide several user required functions. They include: a block-to-block DEC-tape copying program; a program to list the octal contents of any DEC-tape block; a program to modify the contents of any location on any DEC-tape block; a program to translate "DECUS" manuscripts to SCPSYS manuscripts; and DEC-tape read/write subroutines within the bootstrap loader area always available for user programs.

Classroom Application

The course in which SCPSYS was used was Electrical and Computer Engineering 350 at Clemson University. Two sections of the course were taught, one with fifteen and the other with sixteen students. The catalogue description of the course is given below:

E & CE 350 Principles of Digital Computer Systems 3 credits (2, 2)

Introduction to machine structure and programming systems. Topics include: general machine organization, information flow within a machine, internal and external data types and structures; data transfers and communication with external devices and interrelation between software and hardware. The various levels of programming systems are considered, but the main emphasis is place on machine languages. Prerequisite: Approval of department.

Although this is a junior-senior level course, it was assumed that students had no prior background in computer organization and assembler language programming. This was a three credit hour course, consisting of two one-hour lectures and a two-hour "workshop" period which could be used at the instructor's discretion for either a laboratory period or for an additional lecture.

The main text for the course was Computer Organization and Programming, by C. William Gear. The prescribed text proved reasonably satisfactory and was followed fairly closely for about a month during which the basic concepts of the course were introduced. However, because of the availability of the PDP-8 computer, it was decided to base a large part of the course on homework assignments to be programmed on this machine. The availability of SCPSYS as a programming aid contributed significantly to the decision to use this approach. In particular, it was felt that SCPSYS would allow the student to interact with the computer at a basic level, without the tedious task of using numerous paper tapes to realize system software.

The process of familiarizing students with SCPSYS was begun with a thorough demonstration of system utilization. After the capabilities of the system had been demonstrated, each student was given a comprehensive manual describing in detail the different aspects of the system and their use. Then successively more difficult homework problems were assigned to be worked using SCPSYS on the PDP-8. The first assignments were taken from Introduction to Programming published by Digital Equipment Corporation. Many aspects of assembly language programming were included, such as subroutines, I/O programming, and interrupt programming. The final assignment involved determining the sample mean and variance of a block of 100 stored data words. This assignment required knowledge of normalizing, scaling, exponent formation, and I/O operations, in addition to basic arithmetic and control programming.

Although the PDP-8 computer was generally available for course use, it did have other commitments. For this reason the following regulations were established for system usage:

1. Two hours were reserved each weekday for class use. Students could sign up in advance for single hours of computer time during these 2-hour periods.

2. Students could use the computer at any other time when it was not busy. One hour was the maximum time any student could use the machine if there were other students waiting.
3. Several students working together could sign up for longer blocks of computer time if needed. However, individual work was strongly encouraged if time permitted. Students were allowed to use the system during evenings, weekends, and holidays, whenever possible.

From the instructor's point of view, the major advantages of SCPSYS as a teaching aid were:

1. Ease with which students would call, edit, and restore manuscript programs. In particular, the ability to "see" changes as they were made contributed to the student's development of confidence in interacting with the computer.
2. Ease of assembling and loading programs for execution.
3. Ability of the system to hold a student's interest on fairly complex homework assignments involving assembler language programming. Almost all students exhibited some degree of fascination with the system, and this attitude undoubtedly contributed to their motivation and perseverance in completing the homework assignments.

Although overall educational effectiveness of SCPSYS was very good, several disadvantages were noted. These are discussed below, along with proposed methods for overcoming them.

1. The ability for a careless or erratic student to "bomb" the DEC-tape containing SCPSYS as well as other students' filed programs was a principal disadvantage. This was due in part to the impracticality of having a supervisor present at all times. This problem could essentially be eliminated by requiring each student or small group of students to purchase DEC-tapes.
2. The length of time required to obtain hard copy of stored programs was another problem. Especially for large programs, the inordinately long time required to obtain a listing from the teletype resulted in inefficient use of the computer. One way to eliminate this problem would be to add a small line printer to the system. In fact, a printer of some sort is almost essential if moderately long programs are to be assigned to a large class. However, this will require an additional investment of an estimated \$7,000.

In order to obtain student reaction to using SCPSYS, a questionnaire was prepared and distributed to the class at the termination of the course. This questionnaire consisted mostly of specific questions related to the system, but also included questions by which students were asked to express any opinions, favorable or unfavorable, about SCPSYS. The findings of the questionnaire are summarized below.

In order to consider the results in proper perspective, students were first asked to indicate any previous experience in machine language or assembler language programming. About 17% indicated some previous experience. The average time of system usage for students working alone was 1.8 sessions/week with 1.7 hours/session. Also, each student averaged .8 session/week with .8 hour/session working in a group. This indicated a total average of 3.6 hours/week. It is felt that this estimate is high, due to the fact that the questionnaire was distributed at the end of the course when many students were spending considerable computer time completing work which should have been finished much earlier in the semester.

One of the most important questions that arises from using SCPSYS in a computer course is whether the time required to become proficient with the system is worth the benefits gained from using the system. Ninety-two percent indicated that they felt that the time spent to learn the system was worthwhile. Also, the average time to learn how to use SCPSYS was 2.7 hours.

Most students listed the capability of editing with visual aid as the principal system advantage. Other advantages listed were wide variety of system commands, quickness of system response, fast checking of manuscript programs, and general ease of executing programs.

Most system disadvantages listed by students were related to the slowness of the teletype in producing listings. This not only prolonged the time required of each student to complete his homework, but also caused significant waiting times for other students who needed to use the machine. Some students complained about having their filed programs accidentally destroyed by other users.

Overall, most students indicated a fairly strong approval of SCPSYS as an educational aid. In fact, 96% indicated that the use of SCPSYS made the homework more enjoyable, and 88% said that the course as a whole was more enjoyable due to the use of this system.

Conclusions

SCPSYS, a scope-based system for editing, filing, assembling, and executing programs on a 4K PDP-8 computer, has been used in our introductory course in computer organization and programming in the Department of Electrical and Computer Engineering at Clemson University. It has been found that this system not only permits the computer to be used more efficiently, but also that it is conducive to the development of a high degree of man-machine interface. A student questionnaire strongly supported these conclusions. Students were able to become proficient with SCPSYS in a short time, and felt that the time required to learn the system was well worth the advantages the system provided. Because of these overall advantages provided by SCPSYS, a "hands-on" experience on a real digital computer was provided in this course to a significant number of students. In summary, it was generally felt that SCPSYS contributed appreciably to the overall effectiveness of this introductory computer course.

REFERENCES

- Gear, C. William: Computer Organization and Programming; McGraw-Hill; New York, 1969.
- Introduction to Programming: Digital Equipment Corporation; Maynard, Mass., 1970.
- Lovell, Bernard W.: A Simulated Mini-Computer Package to Teach Introductory Machine and Assembler Language Programming; Purdue 1971 Sym. on Appl. of Comp. to Electr. Engr. Educ.; Purdue University; Lafayette, Indiana; April 26-28, 1971.
- Burger, Peter: A Teaching Oriented System for a Mini-Computer; Purdue 1971 Sym. on Appl. of Comp. to Electr. Engr. Educ.; Purdue University; Lafayette, Indiana; April 26-28, 1971.

COMPUTER-ASSISTED NUMERICAL CONTROL PART PROGRAMMING

Paul T. Moriarty
New York City Community College
Voorhees Campus
New York, New York 10036
Telephone: (212) 563-1370

Introduction

In Spring 1971, Voorhees Technical Institute (now the Voorhees Campus of New York City Community College) offered two courses in Numerical Control (N/C) in the Evening Division. The first course, MT 420 (Table 1), dealt specifically with the manual part programming of a drilling and milling machine. Its objectives were to familiarize the students with the principles and theory of N/C, and to give them hands-on experience in the machining of parts by numerical control.

The second course in the sequence, MT 440 (Table 2), had as its objectives, giving the student an exposure to computers and showing them the benefits of computer-assisted N/C, through actual hands-on use of a computer in part programming.

Hardware and Facilities

The Machine Tool Technology laboratory has a Bridgeport 3-axis miller with a Slo-Syn Control Unit, along with a Friden Flexowriter for manual tape preparation and all the tooling necessary to maintain the courses. The Computer Science laboratory is equipped with an IBM 1130 3K CPU, card read/punch, line printer, disk drive, plotter, and paper tape I/O along with six keypunches.

Preparations

I was contacted in October 1970 regarding the Machine Tool Department's desire to interface the two departments and to develop the MT 440 N/C course. I assumed responsibility for this interdisciplinary project. It was obvious that I had to quickly learn at least the following: what is involved in numerical control machining and how computer-assist is presently used and the details and operations of both the Bridgeport miller and Slo-Syn Control unit.

To prepare for the study which followed, I had conferences with the machine tool staff and observed laboratory sessions, I visited the Superior Electric Company, manufacturers of the Slo-Syn system and the Bridgeport Machine Tool Company; I attended the MT 420 course and obtained all the literature readily available concerning computer-assisted N/C.

The Course Approach

One of the course objectives was to give students entire hands-on experience in the use of the computer. Gradual exposure, through the practical application of the computer in increasing degrees of complexity allowed the student to advance slowly to full utilization of hardware and software available for numerical control. This success was accomplished through the simplified design of the software, and having hardware that encourages hands-on usage.

Another objective was to have students program parts with geometric patterns, tedious to do by hand, but made relatively simple with computer-assist. The students' first manual part programming assignment gave them an understanding of the difficulty of advanced programming, and with the use of the computer they were able to program a final part which would have been nearly impossible to do manually.

The Software Design

Soon after acquiring enough information about the N/C equipment and programming, I began to search for 1130 N/C software. Although there are two Type IV programs in the IBM 1130 Catalog of Programs, neither had a postprocessor for the Slo-Syn system. Therefore, I decided to develop "customized" software tailored specifically to the hardware available. Many restraints were considered in the software design: the programs had to be written quickly and with minimum

MT420 BASIC NUMERICAL CONTROL (N/C) MACHINING COURSE
2 EVENINGS PER WEEK, 3 HOUR SESSIONS 7½ WEEKS (45 HOURS)

TOPICAL OUTLINE

- | | |
|--|---|
| <p>I. INTRODUCTION</p> <ol style="list-style-type: none">1. TYPE OF N/C SYSTEMS2. REFERENCE POINT SYSTEMS3. TAPE FORMATS4. FILM <p>II. THE SLO-SYN SYSTEMS</p> <ol style="list-style-type: none">1. THE SLO-SYN MCU CONTROLS2. THE BRIDGEPORT CONTROLS3. SPINDLE-WIZARD CONTROLS4. DEMONSTRATION <p>III. PROGRAM WRITING</p> <ol style="list-style-type: none">1. POINT-TO-POINT2. CODING SHEETS3. TAPE FORMATS4. MISCELLANEOUS FUNCTIONS5. DRILLING DEMONSTRATION | <p>IV. USE OF FLEXOWRITER</p> <ol style="list-style-type: none">1. SET-UP OF MACHINE2. TAB SETTINGS3. CORRECTING ERRORS4. REPRODUCING TAPES <p>V. PROBLEMS</p> <ol style="list-style-type: none">1. DRILLING ONLY2. MILLING ONLY3. DRILL/MILL COMBINED4. Z-AXIS PROGRAMMING5. ROTARY TABLE PROGRAMMING <p>VI. INTERPOLATION</p> <ol style="list-style-type: none">1. CIRCULAR AND LINEAR2. POINT-TO-POINT ARCS3. CONTINUOUS PATH PROGRAMMING4. CONTOURING PROGRAMMING |
|--|---|

Table 1

MT440 BASIC COMPUTER-ASSISTED NUMERICAL CONTROL
2 EVENINGS PER WEEK, 4 HOUR SESSIONS 7½ WEEKS (60 HOURS)

TOPICAL OUTLINE

- | | |
|--|---|
| <p>I. REVIEW OF N/C</p> <ol style="list-style-type: none">1. TAPE FORMATS2. N/C CONTROLS3. COORDINATES4. MANUAL PROGRAMMING5. BOLT-HOLE-CIRCLE PROGRAM <p>II. INTRODUCTION TO COMPUTERS</p> <ol style="list-style-type: none">1. COMPUTER SYSTEM2. COMPUTERS IN N/C3. PUNCHED CARDS AND OTHER MEDIA4. DEMONSTRATION OF SSNCs AND COMPUTER UNITS <p>III. PREPARATION OF COMPUTER INPUT</p> <ol style="list-style-type: none">1. CARD COLUMNS, FIELDS2. USING THE KEYPUNCH3. SSNCs CODING SHEET4. KEYPUNCHING BOLT-HOLE-CIRCLE <p>IV. COMPUTER LABORATORY</p> <ol style="list-style-type: none">1. SLO-SYN PROGRAM2. KEYPUNCHING DATA3. INPUT TO SSNCs4. TAPE LAYOUT FROM SSNCs | <p>V. MACHINE TOOL LABORATORY</p> <ol style="list-style-type: none">1. REVIEW OF CONTROLS2. DRILLING BOLT-HOLE-CIRCLE3. REVIEW OF TOOL CHANGES <p>VI. IDLER-PLATE DRIVE LINK</p> <ol style="list-style-type: none">1. USE OF SSMAC2. CENTER DRILLING ONLY3. USE OF SSNCs4. ANALYSIS OF OUTPUT FROM SSNCs5. HOW TO DEBUG PROGRAMS <p>VII. MILLING OPERATION</p> <ol style="list-style-type: none">1. RESULTS OF EXPERIMENTATION2. ACCURACY OF SSMAC3. TOOL CHANGES AND PROGRAM MODIFICATION <p>VIII. TOTAL COMPUTER ASSIST</p> <ol style="list-style-type: none">1. CARDS FROM SSMAC2. MERGING CARD DECKS3. ACCURACY OF PLOTTER4. DEBUGGING AND ERRORS <p>IX. MACHINE TOOL LABORATORY</p> <ol style="list-style-type: none">1. MILL/DRILL CONSIDERATIONS2. MACHINING OF FINAL PART |
|--|---|

Table 2

testing, due to the lack of time; extensive error-checking and diagnostics were necessary in a student environment; and most important - the system must be interactive with the student to facilitate hands-on usage. The software written was named SSNCS/1130 - Slo-Syn Numerical Control System/IBM 1130; Figure 1 is an overall flowchart of the system, and more important features are listed in Table 3.

For geometric part programming, a separate system was designed, based on Digital Equipment Corporation's "Quickpoint - 8" N/C system. An interested senior computer science student was assigned the task of writing the programs as a term project for one of his courses. The system implemented was labelled SSMAC - Slo-Syn MACros. The term "macros" was chosen since the programs would generate many data blocks from the one line of parameters input to them. The SSMAC system interfaces with SSNCS by punching the data blocks into cards in the format required by SSNCS. The patterns provided by SSMAC are: Bolt hole circle (BHC); line at an angle (LAA); grid (GRD); increment along a line (INC); and points along an arc (ARC). It would have been more desirable to have the macro commands as part of SSNCS itself, but time did not permit their incorporation into the system. It is expected that this may be done later.

Because of the haste with which SSNCS and SSMAC were written and the lack of time for complete testing, students were advised to report any problems or errors they thought they had found. As errors occurred, an error log was kept, and they were corrected as soon as possible. Fortunately, all errors discovered were minor, and corrected by the next meeting of the class. Suggested improvement and modifications were done quickly if of a minor nature, but major modifications were logged for future incorporation.

Course Assignments

The part programs done by the students were structured so that each assignment required the student to grasp new concepts piece-by-piece and to allow him to use the concepts in practice before advancing to new material.

The first assignment was the complete manual calculation, programming and tape preparation for a nine-hole bolt-hole-circle. This gave the student a realization of the difficulty of geometric calculations and a review of manual part programming. Having done this, the next step was the use of the computer in doing the calculations, yet still requiring the manual tape preparation. A program in the Slo-Syn Handbook was used, which read one data card containing the parameters for the circle, calculated, and printed the X and Y coordinates. The student was instructed as to the use of the keypunch for preparation of the parameter card and two control cards for executing the program. In the computer lab they were guided as to the operation of the computer and what buttons to push. Since it was their first experience at the consoles, the task was simple enough to be mastered quickly, giving them a sense of accomplishment, and an easy familiarity with the computer system. The programmed bolt-hole-circle was then machined in the N/C laboratory, providing students with a review of the controls and operations of the Slo-Syn system.

The next part to be done (Figure 2) was taken from an IBM N/C Course Instructor's guide. With this part the student would learn and use the SSMAC and SSNCS systems, along with all the computing hardware. The first step was the center-drilling of all holes, and SSMAC was used for calculation of the two bolt-hole-circles. The cards from SSMAC, along with the manual programming of other points were fed into SSNCS, which checked for errors and provided outputs for the "debugging" of the part program. The students were instructed as to the features and operations of SSMAC and SSNCS and followed the process outlined in Figure 3. This step in the course was the longest because of the large amount of interrelated topics which had to be covered and their complexity. When the center-drilling tape was obtained from SSNCS, the tape was run on the Slo-Syn and the partially completed part was saved for further machining. After programming and center-drilling the idler plate, the student was very familiar with the part, which would be necessary for the operations to follow.

The next step in the programming and machining was the drilling of the holes, which required tool changes for the different drills used. Because the points had been calculated previously, the student concentrated on the use and operation of SSNCS and the 1130. Familiarity and prior experience speeded up the part programming and the machining, despite the added complexity.

Although the Bridgeport was a point-to-point machine, linear and circular interpolation part programming was also covered in MT 420, though not actually used. The possibility of simulating the interpolation by milling an arc with closely-spaced computer-calculated increments had been suggested previously by the Machine Tool Staff. Experimentation was done to see if this were actually possible. Using the ARC routine of SSMAC with one degree angle increments, a tape was produced which machined an arc of reasonable tolerance for student work. This led to the assignment of the final step: milling the perimeter of the idler plate. At this critical accuracy, however, SSMAC developed round-off errors which had to be hand-corrected and

SSNCS FLOWCHART

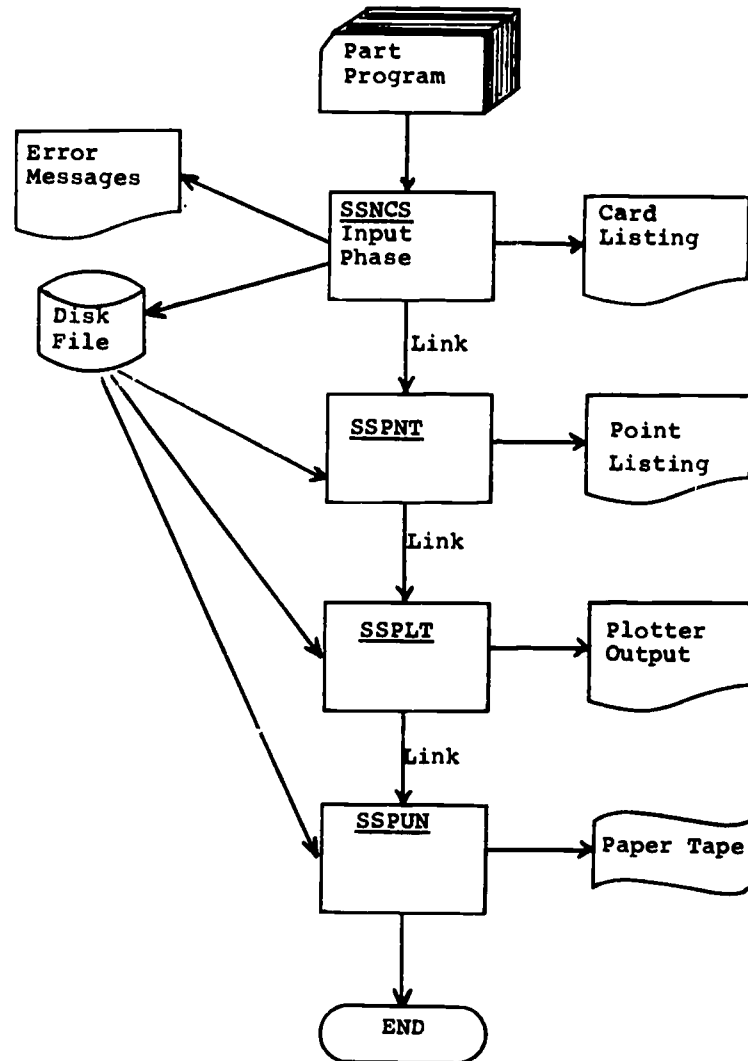


Figure 1

ADVANTAGES AND FEATURES OF SSNCS

1. Card input:
 - (a) preparation of data in standard 80-column cards which are easier to read than paper tape.
 - (b) Easier program modification -- change "unit record" card rather than sequential paper tape.
 - (c) system edits input, prints errors, outputs cards' listing with optional comments for operators.
 - (d) no programmer retraining--cards can be punched from present Slo-Syn coding forms
2. Listings:
 - (a) card listing mentioned above (1c)
 - (b) point listing in incremental steps, incremental inches, and absolute inches from starting position.
 - (c) return to zero (starting position) checked by system.
3. Plotter output:
 - (a) draws all points, notes tool changes
 - (b) plots tool path
 - (c) draws drilling holes to actual size
 - (d) entire drawing drawn to actual size, within accuracy of the plotter (on IBM 1627-- ± 0.01 in.)
4. Punched tape:
 - (a) tape punched with the speed and accuracy of the computer.
 - (b) tape directional arrows are punched into tape, along with an operator-readable label and a label for listing on a Friden Flexowriter.
 - (c) production of multiple copies of a given program tape for back-up or multiple-machine purposes.

Table 3

PART PROGRAMMING FLOWCHART

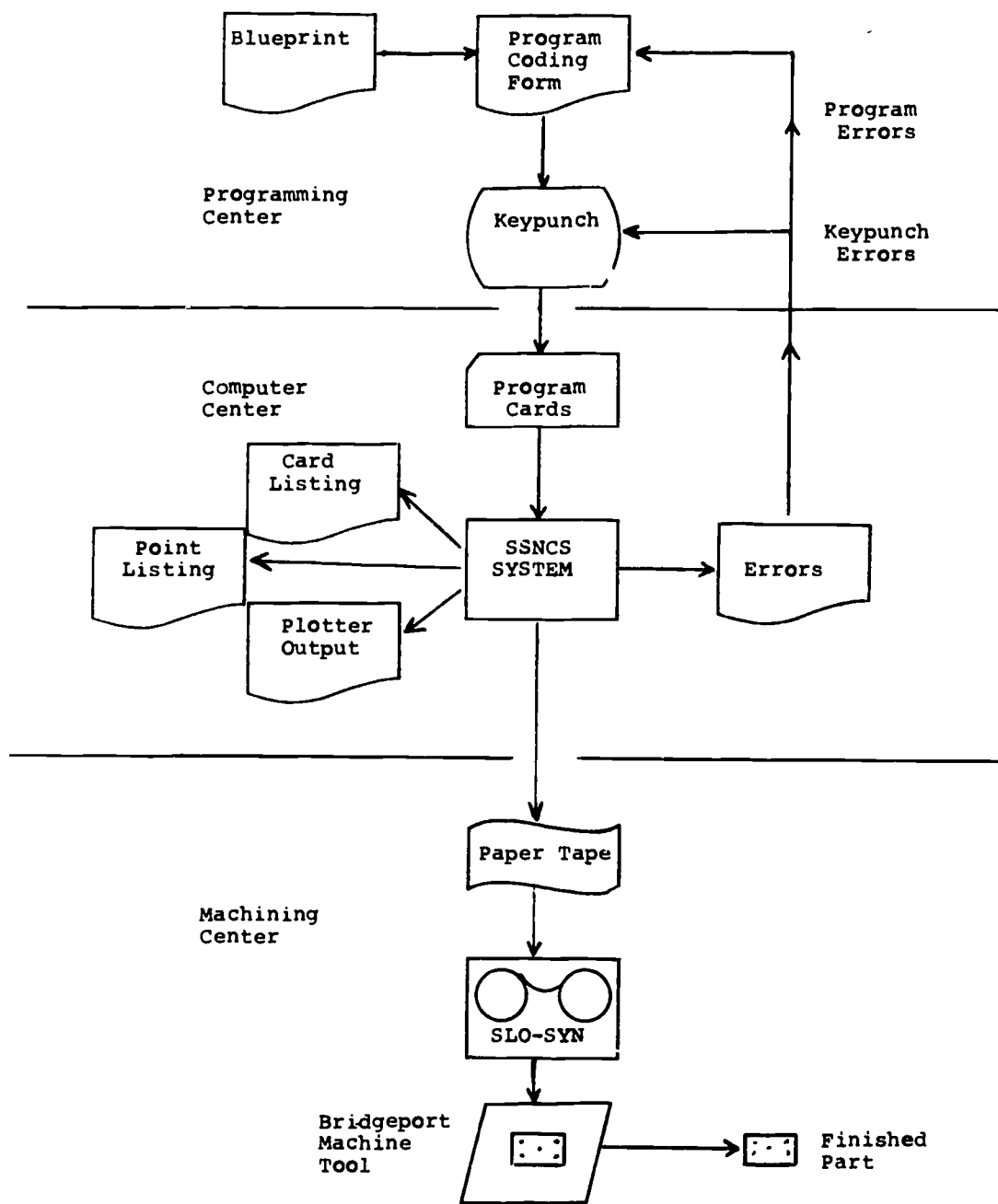


Figure 3

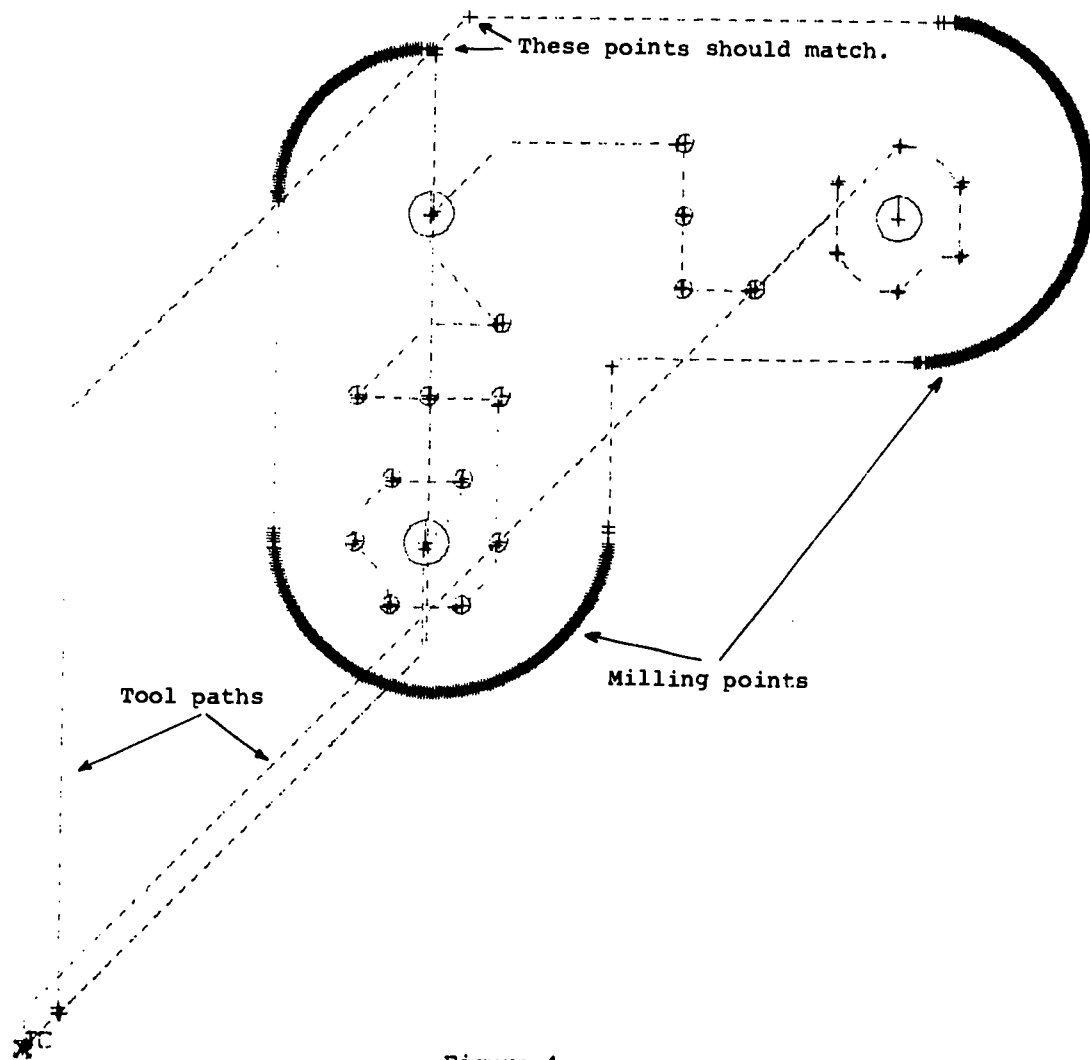


Figure 4

another session of debugging began with SSNCS. The IBM 1627 plotter used has an accuracy of .01 inch, but the number of increments and the accuracy required by the machine tool (.001 inch) resulted in round-off errors, and the milling path drawn was almost one-half inch off the mark (Figure 4). At first this seemed to be a programming error, but a machining run proved the tape correct. Experimentation is being done to resolve these problems. The final machining of the idler plate in every case was successful. Without computer assist this final assignment would not have been possible: program card decks were approximately 550 cards and the tape for the Slo-Syn was 42.25 feet long. With a continuous path system, this volume would be diminished to the point whereby manual programming would be a simple exercise.

Conclusion

Students learned and experienced the entire computer-assisted part programming process, it was entirely "hands-on:" coding, input preparation, running the computer, checking their output, and finally machining the parts on the Slo-Syn. They also used the computer at various "levels" of computer-assist, from simple one-shot programs to the complete hardware and software systems.

Programs were prepared much faster. Input could be prepared simultaneously by six students rather than waiting for one student to finish with the single Flexowriter. Errors both in keypunching and programming logic were corrected much more quickly and eliminated before they were run on the machine tool, which may have resulted in damage.

The design of software with simple input requirements, complete diagnostics, and clear operating instructions, coupled with a computer system that is easy to operate not only allowed the students to run programs more quickly and with hands-on, but also enabled them to clearly experience, to a limited extent, what a computer is in terms of hardware and what it can do for numerical control.

AN ON-LINE MINICOMPUTER IN THE NUCLEAR ENGINEERING CLASSROOM*

Don S. Harner and
W. Waverly Graham, III
Georgia Institute of Technology
Atlanta, Georgia 30332

Introduction

An instructional system has been developed and used in classroom presentations at Georgia Tech in nuclear physics, nuclear engineering, and in a joint course with Emory University in experimental physiology. The computer applications to instruction fall into four categories:

1. Simulation or illustration of physical phenomena by animation
2. Graphical representation of the behavior of mathematical functions as boundary conditions and parameter changes
3. Computation tasks which accelerate solutions to demonstration problems
4. Interactive questions and answers (Computer-Aided Instruction).

Experience with these classroom uses has been good. They have been especially effective in illustrating dynamic processes which are difficult to visualize (categories 1 and 2 above) and in shortening the time required to explain an instructional point (categories 3 and 4 above).

The System Hardware

Elements of the system used in the classroom are shown in the schematic of Figure 1. The computer is a Digital Equipment Corp. PDP-8/I equipped with an 8K memory and 65K disk. A Tektronix, Inc., type 4501 Scan Converter unit which operates in conjunction with the memory scope display-controller, DEC type 34D (modified), is located at the computer. This Scan Converter is essentially a storage oscilloscope with a television raster readout; it provides video plus an RF channel 3 signal which can drive an auxiliary video monitor or standard TV set. This unit's outputs and the computer console teletype lines are connected via a patch panel to a classroom in the same building. In the classroom, a large screen TV monitor and a teletype equipped with a Typoprojector[1] permit student viewing of all control console messages between the instructor and the computer as they are projected onto a large screen along with the graphical output displayed on the TV screen. In essence, one has a computer under instructor control with enlarged teletype and memory scope displays.

Applications

With a powerful graphic display capability, classroom use of the on-line computer for simulation and animation has been the most extensive application. In each of the examples of this type to be described, some nuclear phenomenon of microscopic proportions has been given a dynamic representation in order to improve student perception of the important features of each event.

1. A neutron slowing down and diffusion simulation represents monoenergetic neutrons being injected, one at a time, into a moderating medium whose mass is specified by the instructor during the course of the demonstration. A random number generator determines the initial direction of the neutron and its point of collision with a moderator nucleus. The system absorption and scattering characteristics are determined by statements incorporated in the compact FOCAL[2] program reproduced in Figure 2. As the neutron moves through simulated space, its position is recorded at unit time intervals. Scattering collisions alter its direction and energy and the relative proximity of position points are faithful representations of the particle's speed, i.e., the closer the plotted points, the slower the neutron. Absorption stops the motion representation and terminates the track presentation. See Figure 3 as an example of a typical neutron history track. This demonstration is normally used throughout a one-hour classroom period in which different graphic examples of random neutron behavior are employed as the motivation for an instructor discussion on the important parameters of this process, one of central significance in reactor physics. At one point the randomness of events may be emphasized; at another the equations describing the phenomenon may be displayed by a listing of the program itself. As is evident from Figure 2, the startling

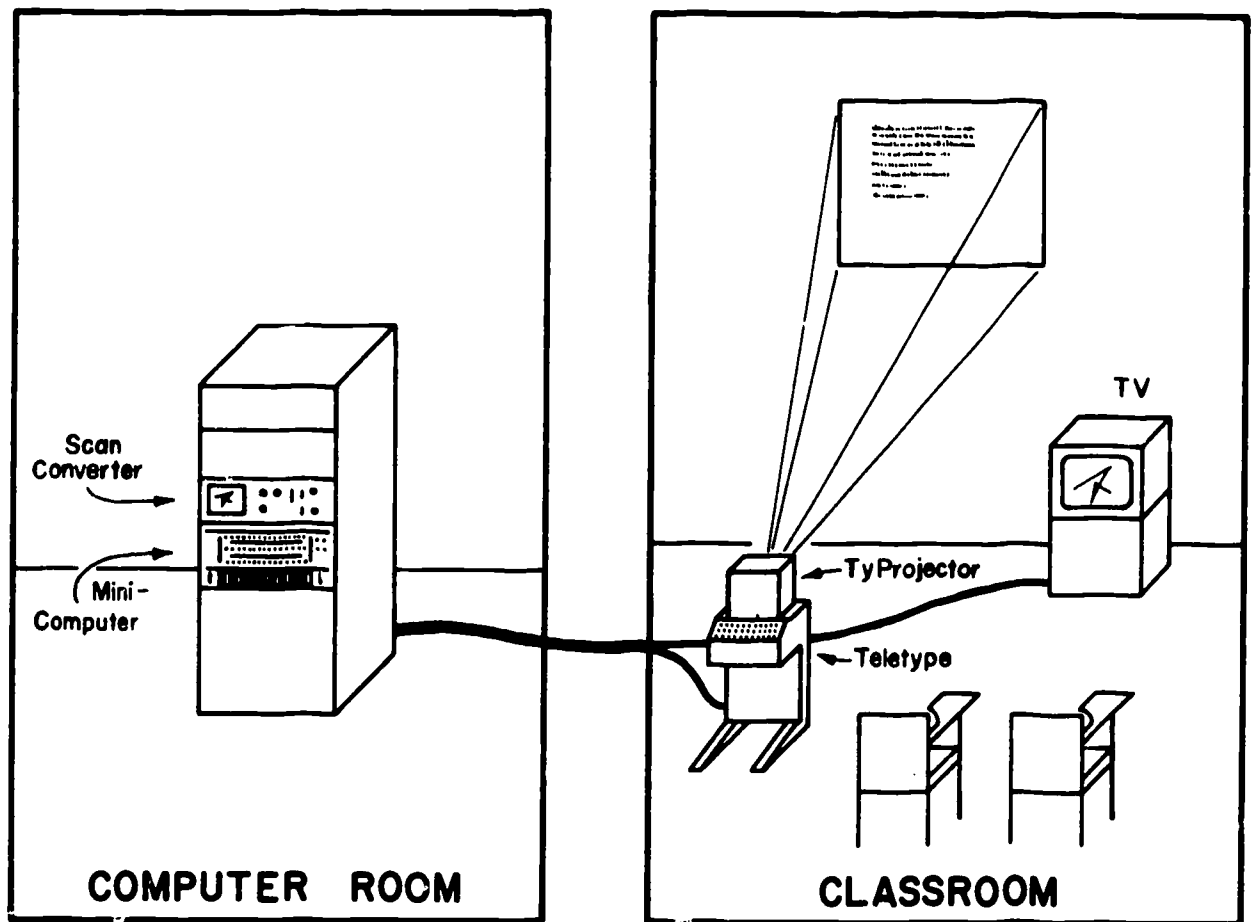


Figure 1. A Schematic View of the "Mini-computer" - Classroom Graphics Arrangement for the School of Nuclear Engineering at Georgia Tech.

W
C FOCAL F 11/11/71

```
01.01 C NEUTRON DIFFUSION IN MODERATING MEDIA; DAVID DIXON, NE705H
01.02 TYPE "THE MASS OF THE MODERATOR IS ";ASK M
01.03 SET P=6;SET PI=3.14259;IF (FX(2,FX(10,6105)));DO 2.4;C STORE MODL
01.05 SET XO=500; SET YO=500; SET PHI=2*PI*FRAN(); SET V=55;SET RR=1
01.07 SET V=V*RR;IF (V-3) 2.3;SET NUM=-P*FLOG(1-FRAN())
01.08 FOR I=0,V,NUM*V; DO 2
01.10 SET CHI=2*PI*FRAN();SET PHI=PHI-CHI
01.13 SET RR=FABS((FCOS(CHI)-(FSQT(FCOS(CHI)2+M2-1)))/(1+M))
01.15 T ""
01.20 GOTO 1.07

02.10 SET XO=XO+FCOS<PHI>*V; SET YO=YO+FSIN(PHI)*V
02.20 SET W FDIS(XO,YO)
02.25 IF (1000-YO) 2.4;IF (1000-XO) 2.4;IF (YO) 2.4;IF (XO) 2.4;RET
02.30 F I=10,20,100;I (FDIS(XO+I,YO)+FDIS(XO-I,YO));D 2.35;D 2.36
02.35 I (FDIS(XO,YO+I)+FDIS(XO,YO-I)+FDIS(XO+I,YO+I))
02.36 I (FDIS(XO-I,YO+I)+FDIS(XO-I,YO-I)+FDIS(XO+I,YO-I))
02.40 IF (FX(2,FX(10,6102)));GOTO 1.05;C ERASE SCOPE
02.50 GOTO 1.05
```

Figure 2. A Programming Example - Neutron Diffusion Simulation.

This FOCAL (TM) program is designed to illustrate graphically the step-by-step history of the process of a neutron diffusing, slowing down, and finally being absorbed in a two-dimensional homogenous media.

This program was student developed as a "homework" assignment and is a "Monte Carlo" solution to the equations for probability of an elastic collision and the resulting velocity of the colliding particle (lines 1.07, and lines 1.10, 1.13 above).

The function, FDIS, displays a point at (X,Y) position on the oscilloscope screen. For purposes of program illustration, the example has the commands (SET, IF, etc.) spelled out, they could have been abbreviated (S, I, etc.) to reduce the program storage requirements. As written however, this program will run in a 4 K-word computer without auxillary storage.



Figure 3. Dynamic Graphical Illustration of the Track of a Neutron Slowing Down in a Homogenous Media of Atomic Mass 10. The neutron is "born" at point 1 at a fixed speed, its initial direction is chosen randomly. The distance to first collision (at point 2) is calculated from a second random number, based on the known interaction probability as a function of distance (exponential attenuation). On reaching point 2, a scattering direction is chosen randomly, the equations for velocity change for this scattering angle are solved, and a new distance to collision (at 3) is chosen similarly, etc. (The particle velocity being indicated by the distance between points.) At each collision the probability of capture is also determined. In the case above the particle was captured at point 11. If the particle wanders, in its "random walk", out of the area of interest, a new particle is generated. In this manner the student can observe the random in-time behavior of a neutron slowing down in a dynamic fashion.

simplicity of the program encourages on-the-spot modification of the formulas in order to pursue some student-motivated question or to demonstrate the effect of a revised absorber or moderator specification.

2. A program which incorporates a Monte Carlo approach to the understanding of radiation attenuation utilizes an animation method similar to the neutron slowing-down application just described. In this demonstration, one hypothesizes the impingement of a single gamma ray (or other radiation) on some attenuating material whose composition is specified. Using probabilities which are determined by the known interaction properties of radiation of the particular energy and material of the specified type, penetration of the material is calculated and tabulated. A histogram of radiation flux vs. penetration distance into the attenuator develops on the display screen as each succeeding gamma ray is followed. The result is a student "discovery" of the exponential nature of attenuation in what is quite literally a computer experiment. This program is included as Figure 4.
3. The use of different coordinate reference systems is explored in a third simulation involving elastic collisions of particles in the center-of-mass and laboratory systems. Properties of these sometimes confusing, always necessary, alternative reference systems are experienced by changing the masses of target and bombarding nuclei and observing the effect on final velocities. The adage that "seeing is believing" is thus well-applied to the laws of conservation of energy and momentum.

Closely related in form to the examples given above is another computer use in the classroom which we classify as simulation, the visual representation of a mathematical function in order to demonstrate how this function behaves as its parameters are changed. Once again, the simplicity of the FOCAL language makes the function definition and the graphics generation command structuring task a straightforward one. We have used this technique to demonstrate the solution of coupled differential equations as they describe a radioactive decay problem. This simulator demonstrates the nature of the decay process by presenting the number of atoms of up to three different species which are present as a function of time. The instructor chooses a single decay chain for illustration of the basic principle, then specifies two or three-component decay and growth systems as desired to show more advanced concepts, such as equilibrium. A version of this simulator has also been applied to the reactor xenon transient problem.

The third application category which deserves mention is one which might be called "demonstration solution acceleration." This category covers a large number of computational programs whose use in the classroom permits a responsive illustration capability and whose use in the laboratory facilitates arrival at the crucial point of data evaluation without excessive "crank-and-grind" data manipulation of the type students find so discouraging.

In a nuclear engineering course dealing with experimental error analysis, an entire series of more than a dozen different programs has been developed for illustration and data-fitting to numerous statistical distributions which are described in the course text by Philip S. Bevington.[3] An undergraduate laboratory in which the radioactive decay of neutron-activated silver is time-analyzed using gamma ray detectors and a multi-channel analyzer, the data are least-squares fit to a two-component exponential decay curve in order to measure the half-lives of the two silver radioisotopes involved. A two-hour at-home exercise is transformed into a two-minute computer operation, permitting each experimental team to compare their results with published values, discuss differences, and re-run the experiment if necessary, all within a three-hour lab period. Contrary to what some computer critics might claim, this procedure has not resulted in any diminished ability of the students to understand the fitting process. Evaluation of that capability through examinations has shown that the classroom development of both the method and the simple computer-stored FOCAL program, coupled with their enthusiastic experience in collecting data which yield literature-quality values for the half-lives, has quite adequately and almost painlessly indoctrinated them in the method. A similar experience has resulted from the use of our facilities in a pulsed neutron study of the neutron diffusion parameters of water, in which a harmonic analysis of the various decay modes is accomplished by the PDP-8/I computer. Without computer data analysis this rather sophisticated experiment could not be nearly so attractive, especially for our undergraduate laboratories. Yet it is one of the students' favorites because it yields a very real experimental feeling for the time frame over which neutrons exist in a moderating medium.

Along this same line, our PDP-12, with its analog signal input capability, has been used in a Georgia Tech physiology course offered jointly with Emory University. Here students were able to measure the response of a frog nerve to a voltage stimulus directly through the application of appropriately located electrodes. On-line signal averaging permitted a continuous data reduction and refining process, the results of which were displayed in real time on the computer's cathode ray tube.

```

W
C FOCAL F 11/11/71

01.01 C PARTICLE PENETRATION-MEAN FREE PATH DEMONSTRATION;D.S.HARMER
01.05 ERASE;IF (FX(2,FX(10,6.105)))IF (FX(2,FX(10,6.102)))
01.10 ASK ?L, T ?

02.10 F X=1,10,1020;S Z=FDIS(X,50)+FDIS(0,X);C DRAW COORD

03.10 FOR I=1,T;DO 4

04.10 SET X=-L*FLOG(1-FRAN());FOR J=1,.5,X;IF (FDIS(J*20,10))
04.20 IF (A-49)4.3;SET X=50
04.30 SET K=FITK(X);S H(K)=H(K)+1
04.40 FOR M=K,.2,.7+K;IF (FDIS(M*20,50+H(K)+20))
#

```

Figure 4. A Graphical Display of Particle Attenuation.

In this program a "Monte-Carlo" solution is made for the distance each particle travels before collision. On collision, a histogram of number of collisions at X , within ΔX vs. X is incremented. As the total number of particles examined increases, the exponential nature of this attenuation can be observed in the histogram. The histogram also illustrates clearly the statistical nature of the interaction process, and the approach to true "exponential" behavior, given a sufficiently large number of events.

The last example of computational use which we have made with our small computer is in criticality calculations, a most important area for nuclear engineers. Classroom use of this reactivity model of a nuclear reactor permits the evaluation of a proposed assembly plus a study of its sensitivity to changes in geometry, materials, or chemical concentrations. It is a flexible tool for accelerating student perception of the relative importance of the various parameters in such a system.

The final category of computer application with which we have experience is that of interactive questions and answers. As most commonly used, this refers to a computer aided instructional environment in which only one student interacts at a time. We have had success in a classroom exercise using group CAI with a tutorial program in nuclear systems. The typewriter made the teletype conversation visible to the entire class while the questions and tutorial text were displayed on the video screen. Enthusiastic participation was achieved through highly interactive discussion among the students about the answers to each question posed.

Conclusion

In conclusion, we have had good student response to several types of classroom and laboratory applications of an on-line computer for simulation of physical processes, visualization of mathematical functions, computations which improve the learning to effort ratio, and interactive drill and practice on a group basis. Much of the flexibility of our particular system is related to the simultaneous power and multiplicity of the FOCAL language with its display command structure.

*A portion of the equipment used in this study was supplied under a U. S. Atomic Energy Commission Grant for equipment for support of education in Nuclear Science and Engineering (AEC Grant No. MSE 11-69).

REFERENCES

1. Registered trade mark of Bolt, Beranek, and Newman.
2. Registered trade mark of Digital Equipment Corporation.
3. Philip R. Bevington, Data Reduction and Error Analysis for Physicists, McGraw-Hill Book Company, Inc., New York, New York (1969).

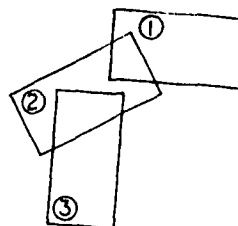
COMPUTER APPLICATIONS TO KINEMATIC SYNTHESIS
OF FOUR BAR MECHANISMS

Richard C. Schubert
Joseph H. Gill
Western Michigan University
Kalamazoo, Michigan
Telephone: (616) 383-4021

The course Mechanism Analysis is included in the Undergraduate Curriculum in Mechanical Engineering Technology at Western Michigan University. It involves both kinematic analysis and synthesis. The analysis is conducted along classical lines whereby velocities and accelerations are solved by analytical and graphical methods. The purpose of this paper is to discuss the kinematic synthesis aspect of the course.

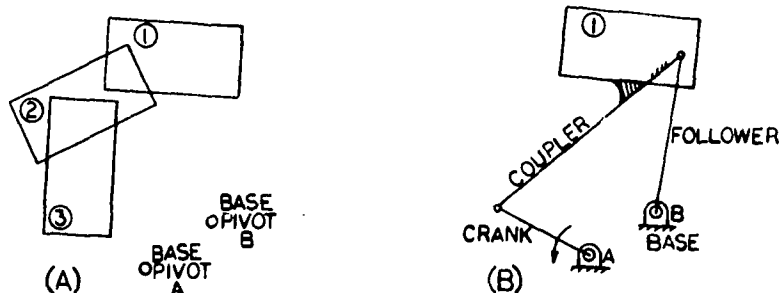
Synthesis, a nebulous word, requires a definition as it relates to the subject matter. In Mechanism Analysis we restrict synthesis to planar mechanisms in general and four-bar mechanisms in specific. For planar four-bar mechanisms there are three types of synthesis: 1. Body Guidance, 2. Path Guidance, and 3. Function Generation. By way of example consider the Body Guidance synthesis problem. Referring to Fig. 1, suppose the body shown is to be guided through the three discrete positions by a four-bar mechanism. Those familiar with kinematic synthesis will recall that there is an infinite number of solutions to this problem. Restricting the location of the two base pivots, as shown in Fig. 2a, reduces the problem to a unique solution. The specific solution is shown in Fig. 2b.

FIGURE 1 BODY IN PLANE MOTION
MOVING THROUGH THREE
DISCRETE POSITIONS



The four-bar mechanism is shown in the first position with the body attached to the coupler. The four-bar mechanism will carry the body through the three prescribed positions. In summary, synthesis requires one to determine the geometry of the mechanism that will generate the prescribed motion of the body. Examples where synthesis is applicable in the automotive design field include hood hinges, door hinges, rear suspension, and carburetor linkages.

FIGURE 2



The student is presented with two approaches, a graphical and analytical method. Initially the problems are designed with the base pivots of the four-bar mechanism fixed. As stated above, this insures a unique solution. The student becomes familiar with the graphical method and subsequently verifies it with an analytical approach. The analytical approach involves the simultaneous solution of three equations involving transcendental functions. At this point the computer becomes the catalyst. At Western Michigan we have the Digital PDP-10 Time Sharing System with approximately 15 consoles available to the engineering student. Students are assigned users' numbers with appropriate time allocations. Rather than ask the student to write the program to solve the equations, a library program is provided. It is only necessary for the student to introduce the data in the proper manner to obtain the solution. The solution is in

the form of x and y coordinates of the moving pivots (see appendix). By assigning problems with unique solutions the student becomes familiar with the operation of the computer - how to access the system, introduce data and obtain answers, and how to terminate use of the computer.

All problems are not uniquely defined. In the discussion above the base pivot locations were specified. Consider the film advancing mechanism of a motion picture projector shown in Fig. 3. Its specific function is to move the film down rapidly (see path 1), release, return, and engage the film again (path 2). This is accomplished with a four-bar mechanism by extending the coupler to include the point describing the path shown. The mechanism is driven by a constant speed motor through link 1. Since the base pivots (A and B) are located on the frame of the projector and could be attached elsewhere, who is to say that this is the best design or even a good one? Fig. 4 illustrates an alternate design that accomplishes the same function. Which is the better design?

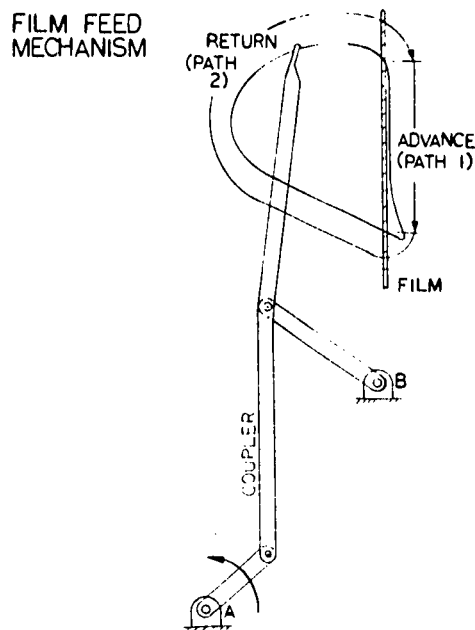


FIGURE 3

FILM FEED MECHANISM
ALTERNATE DESIGN

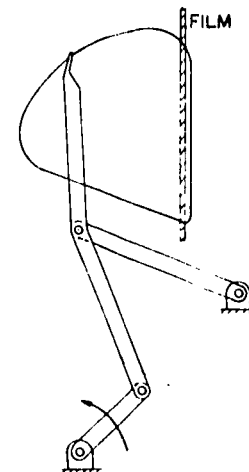


FIGURE 4

The concept of design optimization is introduced to the student in Mechanism Analysis which attempts to answer this question. Three main areas of optimization are considered.

The first considers the relative size of the four links. If any one link is 10 or more times as long as any other link then the mechanism is rejected. The 10:1 ratio is somewhat arbitrary, however, from a practical packaging standpoint a limiting ratio must be observed. Certainly a mechanism with one link 6 inches long and another link 5 feet long would be difficult to package under the hood of a car.

A second criterion considers the transmission angle. This is defined as the smallest angle between the coupler and the driven link (see Fig. 5). The transmission angle will vary from a maximum to a minimum value as the driving link is turned. It is desirable to have the minimum transmission angle as large as possible. For example, if a mechanism with a zero transmission angle stops in this position, it would be theoretically impossible to start again. Furthermore, it has been shown in the literature that wide ranging transmission angles result in large accelerations in high speed operation. It would be desirable to limit the minimum transmission to 40 degrees, although this is not always possible.

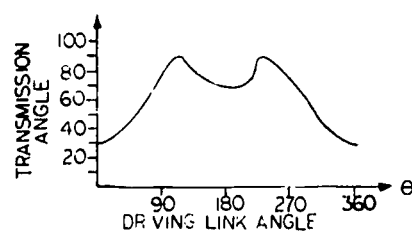
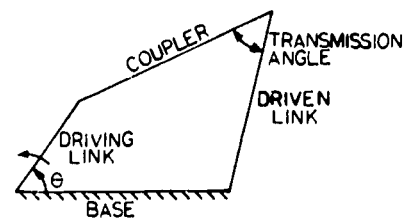


FIGURE 5



The third criterion considers the classification of mechanisms. Some four-bar mechanisms will have no link that can turn through 360 degrees, thus eliminating them from an application with constant speed motor drive. On the other hand some mechanisms need not turn 360 degrees, such as hood hinge mechanisms. Hardings Inequality Method can be applied to a four-bar mechanism to determine if it is the class of mechanism that will or will not rotate through 360 degrees.

Although this comprises only some of the criteria for which a design may be judged it gives the student a basis on which to choose. All of the aforementioned methods have been programmed and included in the computer library. Rather than one computer program to cover all the criteria, each has been programmed separately, thus giving the student the option of using them individually or as subroutine of a general program that he develops.

A term project is assigned to the students in which the motion of a body is to be carried through three positions as shown in Fig. 6. However, in this case, the location of the two base pivots are restricted to area locations. It is observed that for each location of base pivot B there is theoretically an infinite number of possibilities for pivot A. The converse is true when pivot A is fixed. This constitutes a double-infinite number of solutions - a typical real world design problem. Students are given approximately 8 weeks to explore possible combinations to arrive at the best solution based on optimization criteria. Results are compared the final week of class.

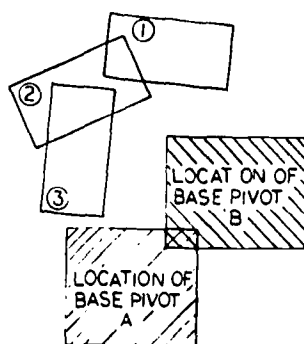


FIGURE 6

In summary, the student familiarized himself with the computer programs after first solving problems by conventional methods, i.e., graphical or mathematical solutions using manual techniques such as slide rule or desk calculators. He then applies these concepts to a term project where the design is virtually open-ended and results must be justified based on optimization criteria. In this way, the student studies the initial results of computer output, analyzes the data, and then resubmits information to refine the design. He thereby controls the destiny of the final design.

APPENDIX

A computer program for body guidance synthesis is written in basic language. It incorporates the input routine rather than the read-data method. This gives the student more flexibility since he can immediately introduce new data contingent on the output of the previous calculation.

Consider a body that is to move through the three positions in Figure A1. The two fixed pivots are located as shown. By setting up a convenient coordinate system with the origin located at one of the pivots, the corresponding x and y coordinate of a particular point on the body is determined ($X_1, Y_1, X_2, Y_2, X_3, Y_3$). The angle of rotation of the body is then determined with counter-clockwise as positive (T_2, T_3). Computer input-output is as follows:

```
Enter X0, Y0
? 0,0
Enter X1, X2, X3
? 1, 2, 3
Enter Y1, Y2, Y3
? 1, 0, 2
Enter T2, T3
? 30, 60
```

```
X - coordinate = -2.2791 [Computer output for moving pivot attached to
Y - coordinate = 0.29975 fixed pivot A.]
```

```
Enter X0, Y0
? 3,0
Enter X1, X2, X3
```

```

? 1, 2, 3
Enter Y1, Y2, Y3
? 1, 0, 2
Enter T2, T3
? 30, 60

```

X - coordinate = 1.1097 [Computer output for moving pivot attached
Y - coordinate = -2.69615 to pivot B.]

The mechanism is shown in the first position in Figure A2 with the body affixed to the coupler.

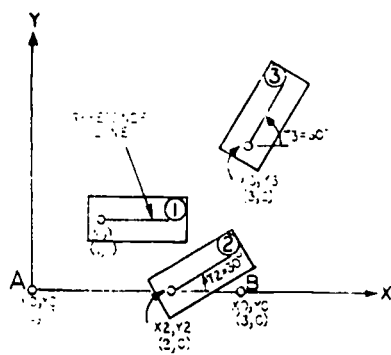


FIGURE A.

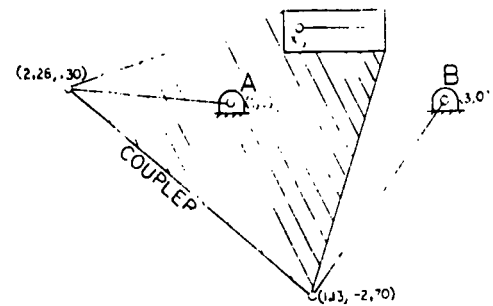


FIGURE A2

AN ANALOG COMPUTER OPTIMIZED FOR UNDERGRADUATE INSTRUCTION

Stephen G. Margolis and Hinrich R. Martens
State University of New York at Buffalo
Buffalo, New York 14214
Telephone: (716) 831-5321

Introduction

In the teaching of courses which introduce the undergraduate student to dynamic, time dependent aspects in the mathematical and physical sciences (i.e., electrical circuits, dynamics, differential equations, etc.), we have always sensed a need to illustrate the associated mathematical equations in a simple and qualitative manner. Considering various possibilities, including movies, videotapes and time-shared digital computer terminals, we have concluded that a simple analog computer capable of repetitive operation would be ideally suited to meet this need. However, commercially available analog computers require some degree of proficiency in programming and assume that the student (and his instructor) have a certain amount of inclination towards electronic hardware. These factors have always created a barrier preventing a fuller utilization of analog computers in introductory mathematics, physics and engineering courses.

In this paper we describe the SUNYAC, a special purpose analog computer which is intended solely for use in the instruction of undergraduate students at the sophomore through senior level. This computer differs from other available analog computers in that it was designed from the ground up to be a tool of instruction, not a tool of research. Consequently, its principal object is not the precise solution of differential equations; rather, its object is to be a tool by which students will gain insight into the structure and meaning of differential equations and into the properties of their solutions.

How does this machine differ from commercially available analog computers? Primarily, it differs by accepting a more restricted set of problems and in exchange offering much greater simplicity of programming and operation. For example, the student wires up his problem using a maximum of eight wired connections. In many cases fewer connections are required; for instance, for a differential equation of the type $dy/dt + ay = \text{constant}$, three connections are necessary: one for the left hand side of the equation; one for the right hand side and a last one for the display oscilloscope. The coefficients are set by a few slide switches and then the solution to a problem is immediately available in real time as a meter deflection, or as an oscilloscope display refreshed forty times per second. Typical setup time for any problem is less than one minute. In contrast, commercially available analog computers have more complex wiring, require that coefficient potentiometers be set by indirect means, and do not always provide oscilloscope display of repetitive solutions. The cost of the SUNYAC is less than \$1,000, including the companion oscilloscope, a price which encourages the availability of enough of these machines for hands-on use by every student.

It is readily granted that this little analog computer trades-off flexibility for simplicity. But in the intended application these losses are more than offset by its low cost and extreme ease of patching and coefficient setting. As a consequence, practically no background or prior preparation in analog computer theory is demanded of student or instructor.

Design of the Machine

The salient design features of the SUNYAC are as follows:

1. Integration and summation are performed by operational amplifiers using field-effect transistor (FET) input stages.
2. Mode switching in both the manual and repetitive modes of operation is performed by complementary symmetry metal-oxide-silicon (CMOS) solid-state switches. Thus no mechanical relays or choppers are needed.
3. Initial conditions are hard wired to the integrators. Initial conditions are restricted to 21 integer values between -10 and +10 volts (including zero); these are set in by four switches using binary-coded-decimal coding in a 1225 sequence. Each integrator may have up to four inputs.
4. Six coefficients are available each of which is step adjustable to 41 values between 0.05 and 2.05 in increments of 0.05. Each coefficient is controlled by six switches and is set using binary coded decimal coding in a 1225 sequence.

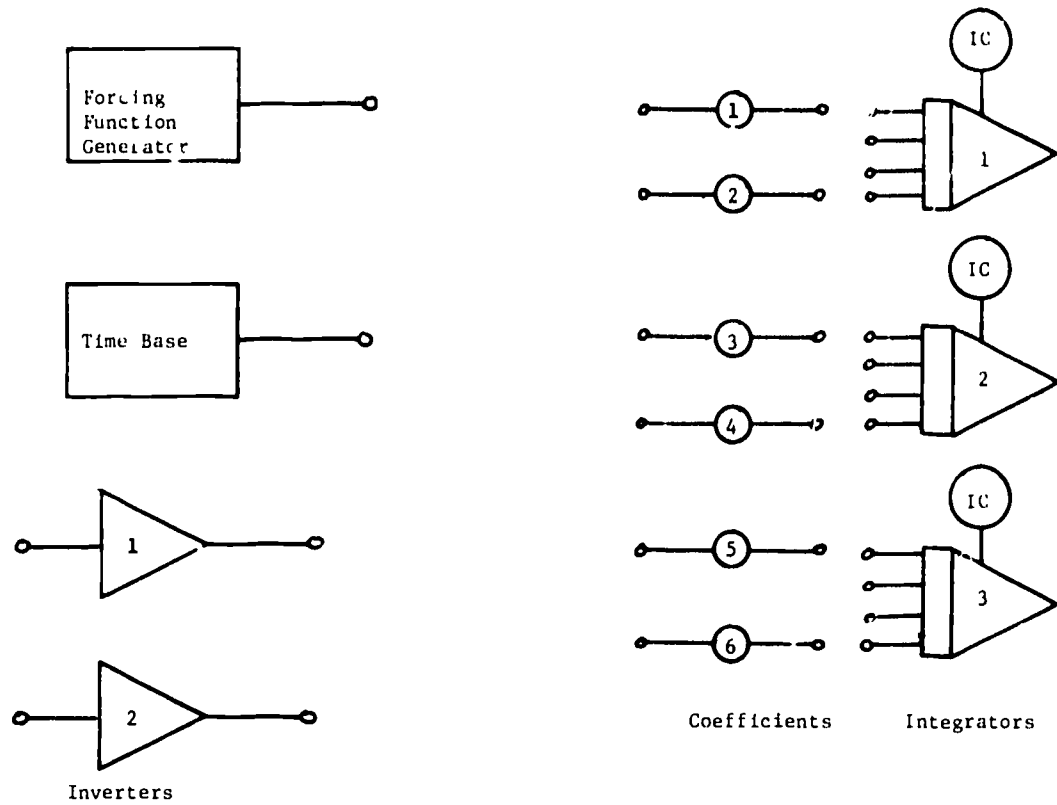


FIGURE 1. Layout of the Computer

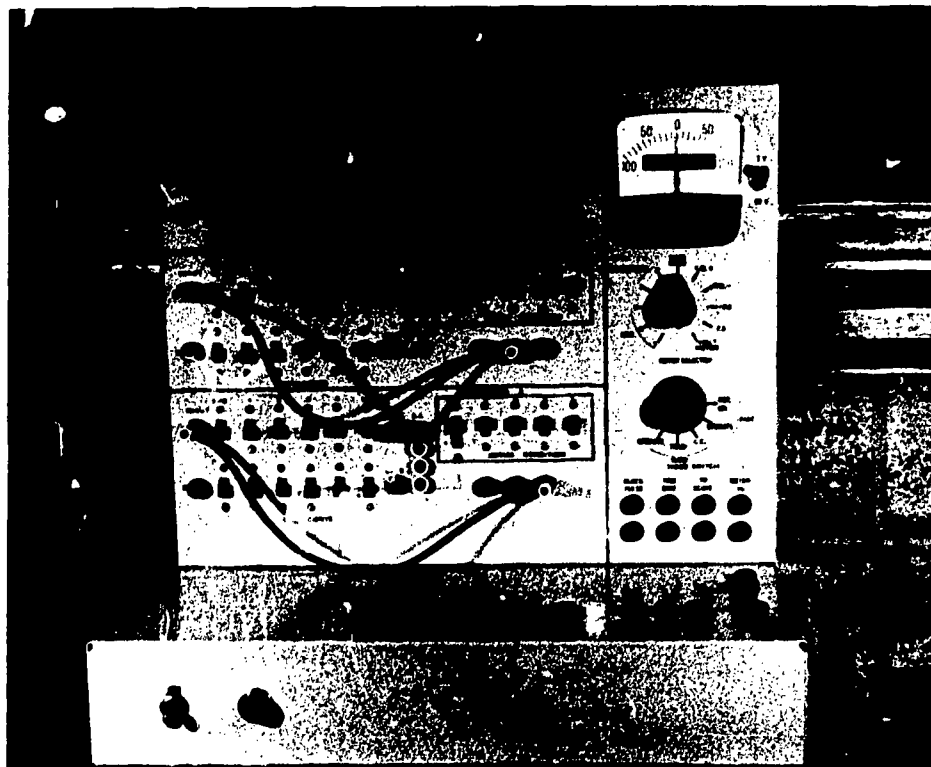


FIGURE 2. SUNYAC Front Panel Layout

Figure 1 is a schematic layout showing the components available in the computer. The computer consists of a forcing function generator, a time base, six adjustable coefficients, three integrators with integral initial condition switches, and two inverters. Normally, each integrator has two coefficients associated with it, but, if necessary, two other coefficients may be borrowed from another integrator. The physical layout of the computer is shown in Figure 2. The coefficient switches are arrayed on the left side of the main panel. The five switches which control the magnitude and size of the initial conditions for each integrator are arrayed down the center of the main panel. Below these switches are the outputs of each integrator. The inputs and outputs for the three inverters are located in the near right hand corner of the lower panel. The controls for the function generator are located along the right side of the lower panel. (Not visible in figure.) In non-repetitive operation, the selector switch located under the meter allows the meter to read the outputs of the three integrators, the two summers, and the function generator. In repetitive operation, the same switch selects the signal fed to the Y-axis of the oscilloscope. The mode switch, located in the right center of the main panel, selects the mode of operation of the computer. In the "initial conditions" mode, initial conditions are set. In "hold" mode, all functions including the time base are frozen, and in "operate" the differential equations are solved. By turning the switch fully counter clockwise, the computer is put into the "repetitive operations" mode in which the time scale is speeded up by a factor of 1,000 and the oscilloscope picture of the solution is refreshed forty times per second.

Figure 3 shows all of the connections and settings which are necessary to solve the set of differential equations:

$$\begin{aligned} \dot{x}_1 &= -x_1 & x_1(0) &= 9. \\ \dot{x}_2 &= -x_2 & x_2(0) &= 5. \\ \dot{x}_3 &= -x_3 & x_3(0) &= 10. \end{aligned}$$

Figure 4 shows the connections and switch settings required to solve the simultaneous differential equations:

$$\begin{aligned} \dot{v} &= -0.5v - 1.5x & v(0) &= 9. \\ \dot{x} &= -v & x(0) &= 5. \end{aligned}$$

Circuit Details

In order to realize the educational objectives of the SUNYAC some novel design features were devised. Of the design requirements, the most prominent was the requirement for rapid refreshing of the oscilloscope display which led to the use of a solid-state integrated-circuit switch in mode switching. In turn, the use of these solid-state switches simplified the overall design and reduced the cost.

The basic unit in the operation of the computer is the integrator. Four identical integrators are used in the computer, three in the solution of equations, and the fourth to generate the time base. The integrator circuit diagram is shown in Figure 5.

Referring to Figure 5, the upper COSMOS switch is closed in the initial condition position, connecting the initial conditions network to the operational amplifier. The two back-to-back silicon diodes provide over-voltage protection for the COSMOS. In the hold position both COSMOS switches are open. In the operate position the "ic" COSMOS is open and the "operate" COSMOS is closed connecting the coefficient network to the summing junction. Again, back-to-back diodes provide protection for the COSMOS switch. The coefficient switches connect the input resistors in parallel. For example, to obtain a coefficient of 0.75, the .5 switch, one of the .2 switches, and the .05 switch would be closed. Note that one end of all of the input resistors is connected to the input terminal and that, depending on the switch position, the other end of each resistor is either connected to the summing junction (a virtual ground) or to ground itself. Thus, as seen from the input, the input resistors all appear to be connected in parallel and represent a constant impedance which turns out to be 0.488 megohms. Because the input impedance is thus fixed, the design of an attenuator to extend the range of the coefficient, if desired, would be very simple.

During repetitive operations, the "ic" and operate COSMOS switches are alternately closed and opened. The wave-forms required to perform this function are generated by an asymmetrical

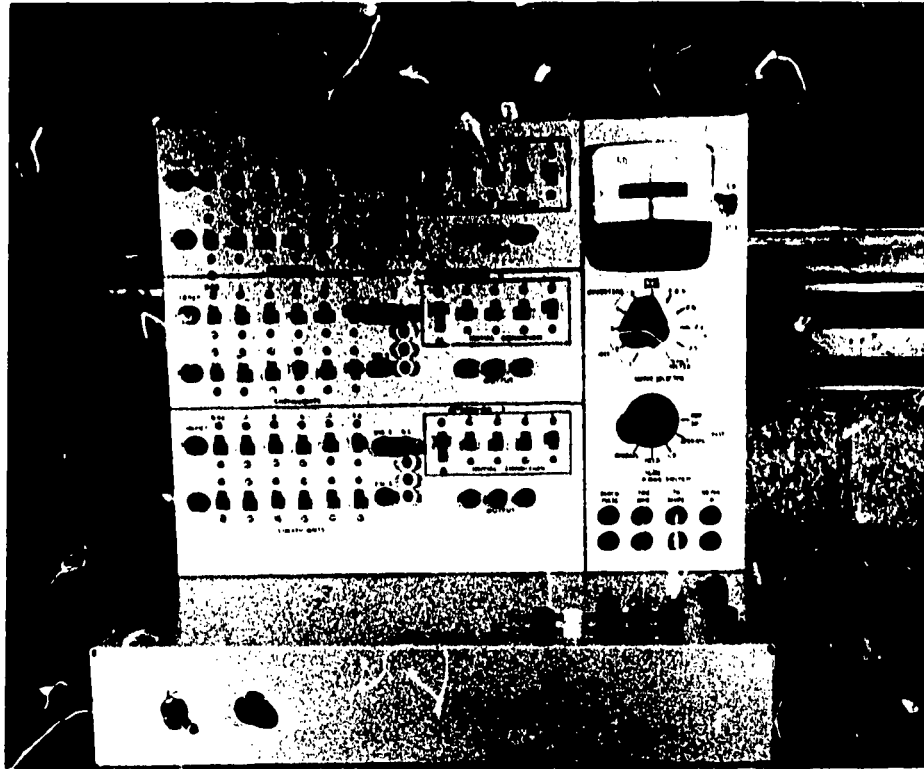


FIGURE 3. Patching for Three Simultaneous Equations

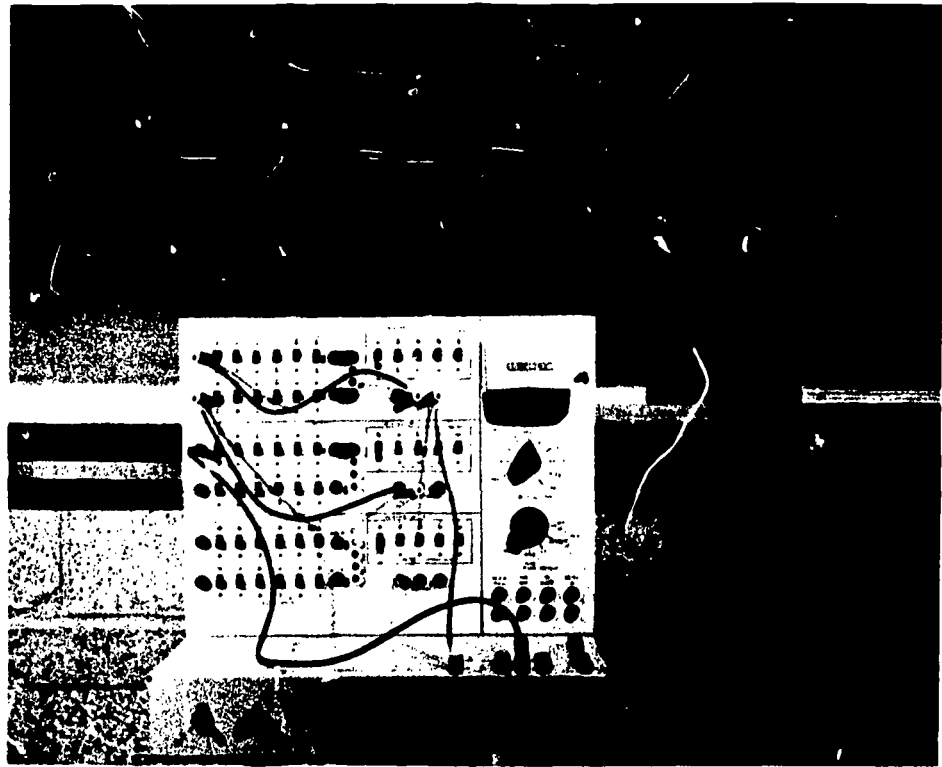


FIGURE 4. Patching for a System of Two Simultaneous Equations

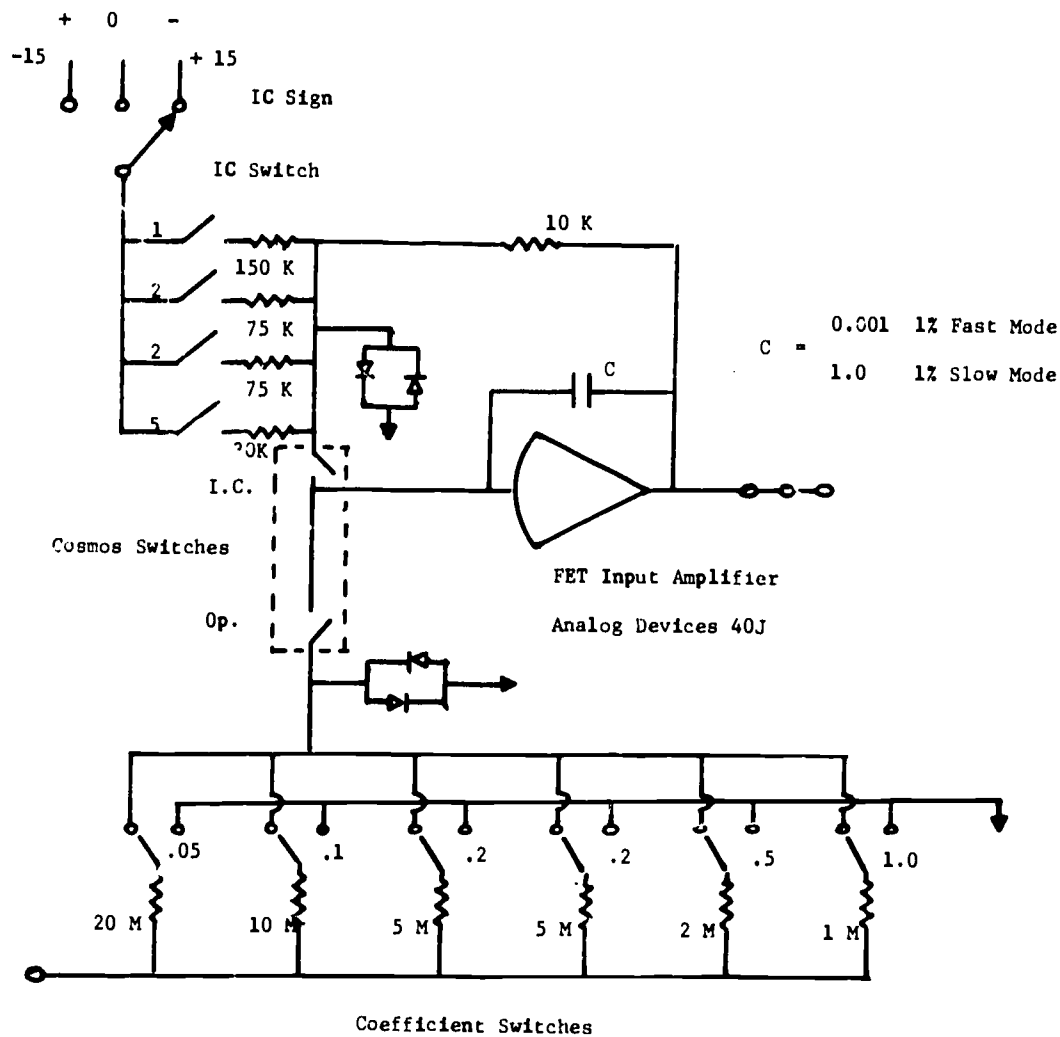


FIGURE 5. Integrator Circuit

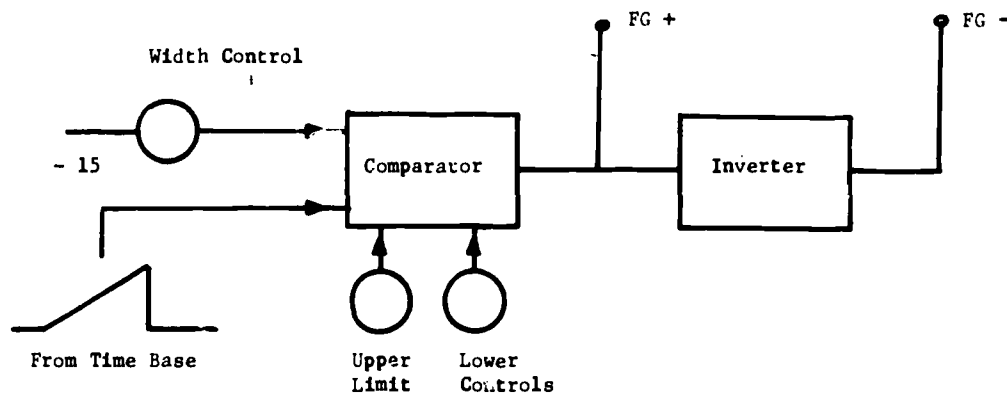


FIGURE 6. Schematic of Function Generator

multivibrator. The circuit used to generate the time base is identical to that used for the other integrators except that its initial condition is fixed as zero and a single input resistor is used to generate a slope of one-half volt per second. Thus, the time base voltage is a triangle wave which starts at zero and reaches ten volts in 20 seconds, in the slow mode.

In the slow mode of operation, the time base may be used to operate an x-y plotter or the horizontal axis of a storage type oscilloscope. In the rep-op mode of operation, the time base is speeded up by a factor of 1000 so that it goes from zero to ten volts in 20 milliseconds.

A block diagram of the function generator is shown in Figure 6 and some typical output waveforms of the function generator are shown in Figure 7. As can be seen in the figure, the function generator can generate pulses, step functions, and one-step staircase functions. With the use of one integrator ramp functions can also be generated. In switching from slow to repetitive operation, the time scale of these functions is automatically speeded up by a factor of 1000, while the amplitude scale is left unchanged. Other functions, such as exponentials and sinusoids can be generated by appropriate connections of the integrators of the main part of the computer.

Applications

The SUNYAC was conceived in the spring of 1971 with a prototype constructed by early summer of 1971. Since then it has been actively tested and a second improved model has been constructed. It has been exposed to a number of different applications which are briefly summarized here. These will testify to the versatility and convenience of the SUNYAC.

a. The prototype model of the computer was used during the summer of 1971 in a workshop on differential equations (sponsored by the National Science Foundation) attended by 27 faculty members from 9 community colleges in New York State. This workshop dealt with a detailed analysis of current practices and new trends in the teaching of elementary differential equations. The SUNYAC was included in this analysis. One of the recommendations emerging from the workshop report consists of encouraging teachers of differential equations to demonstrate the qualitative nature of solutions to differential equations using an analog computer. A partial listing of problems for student solution which were proposed and implemented using the SUNYAC follows:

1. Simple exponential solutions
2. Damped sinusoidal solutions
3. Superposition of solutions
4. Compound interest problems
5. Fluid mixing problems
6. Simple vibration
7. RC, RL, and RLC network problems
8. Demonstration of the impulse as the limit of rectangular and exponential pulses
9. Radio-isotope decay chains

b. During the fall semester 1971 the computer was used by faculty members of SUNY at Buffalo to demonstrate to a junior class state-variable solutions using the concept of the matrix exponential e^{At} . It was also used to demonstrate to a sophomore class in Electrical Engineering the effect of varying the initial conditions of a second-order differential equation, to demonstrate the effects of varying the damping ratio and to illustrate graphically the concepts of underdamped, overdamped, and critically damped solutions.

c. At the present time the second version of the computer is undergoing evaluation at Corning Community College for use in a course in differential equations intended for math majors, science majors and pre-engineering students.

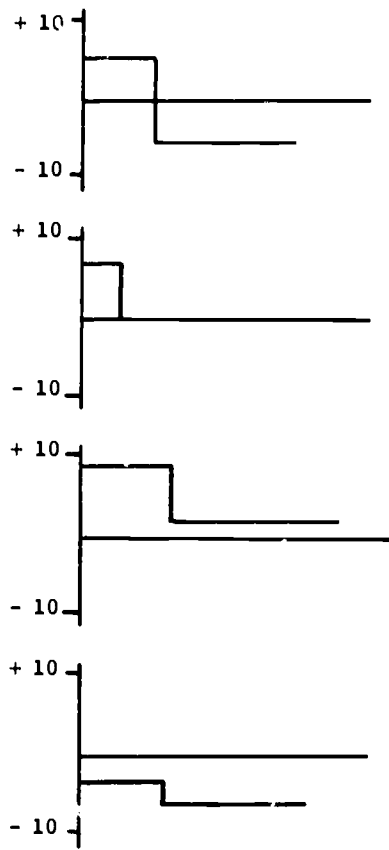
These and other similar applications lead us to conclude that SUNYAC offers no programming barriers to students and instructors who are untrained in the use of analog computers. By virtue of its simple and direct features even a person totally new to analog computing can master its operation within one hour.

A manual complete with operation instructions and fully worked out sample problems has been prepared, reflecting experience to date.

Conclusions

Two models of SUNYAC, a special purpose analog computer optimized for use in instruction, have been designed, built and tested. From the student's point of view, the performance of SUNYAC is equal to that of more expensive and elaborate commercial analog computers, but SUNYAC

Step Functions & Pulses



Ramp Functions Generated with
Function Generator and One Integrator

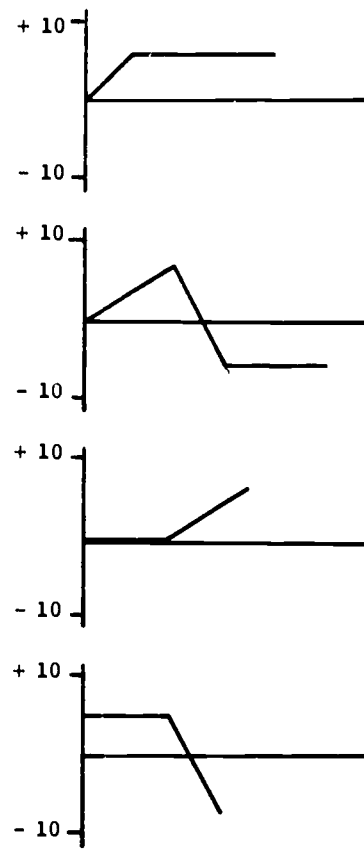


FIGURE 7. Function Generator Waveforms

is much simpler to program and operate. The setting of coefficients and initial conditions is direct and simple. No hardware considerations intervene between a student's thinking about changing a coefficient and his actual implementation of the changed coefficient on the machine. The results of changes in coefficients and initial conditions are immediately apparent to the student on the oscilloscope screen.

Based on use so far, it appears that machines of the SUNYAC type are well suited for second-year courses in differential equations, circuit analysis, and dynamics. In such applications, the machine should prove particularly attractive to two-year colleges in view of their relatively limited budgets and their limited need for research-oriented machines. Because of the low cost of the machines, it should be feasible to provide enough machines to give every student hands-on experience with the machine solution of ordinary differential equations.

For four year colleges, our experience to date with the machine indicates that it has value in teaching sophomore and junior engineering students about the formulation and solution of state equations. With its capability of dealing with third order systems, it can simulate most of the salient properties of linear control systems as taught to seniors in first courses in linear control. Thus we see this as a machine which has applications in courses at the sophomore, junior and senior levels in the undergraduate curriculum in engineering, mathematics, and science.

ACKNOWLEDGEMENT

Two models of the SUNYAC were constructed by Mohammed Motiwala, with the assistance of our technicians A. Longley, W. Willerth and W. Berent. Photographs in this report are by William Margolis. This research was supported in part by NSF grant GY 6593.

SPATIAL MODEL BUILDING IN THE SOCIAL SCIENCES

Philip M. Lankford
University of California
Los Angeles, California 90024
Telephone: (213) 825-1071

Although the use of computers has now become an accepted tool of social sciences research in geography, the impact upon undergraduate education has been slight. If the student is exposed to computers, it is usually not until a quantitative methods course late in college. This paper reports the great success of using the computer in a freshman class as a tool in spatial model building.

The Class

Geography 1C, Introduction to Locational Analysis, is the final course of the three quarter sequence required of all geography majors. The course is open to freshmen, but since very few students enter college as geography majors, many of the students are juniors or seniors. Lectures introduce the students to central place theory, agricultural location theory, industrial location theory, interaction models, and urban-regional growth models. The laboratory part of the class consists solely of computer exercises in spatial model building.

The General Problem

The general assignment for the quarter is to develop and test a spatial model. The student is to choose a current problem of social concern, such as poverty, as a topic of research. Using the theories and models developed in the lectures, the student deduces statements about the spatial components of the problem. After checking census materials, testable hypotheses are developed about a specific variable and its relationship to several other variables. The chosen variables are coded onto eighty column sheets. Observations are recorded for a contiguous group of counties or tracts, or for cities within a region, depending upon the particular problem. After the data are keypunched and corrected, the specific hypotheses are tested with a simple, two-variable regression program, QIKREG. The student then modifies his simple descriptive model based on the results. A second assignment produces, with the SYMAP program, contour maps of the original variable being "explained" and residuals from the significant regressions. The student then prepares a final report, drawing conclusions about his spatial model and commenting on the difficulties encountered.

Methodology

The first point of the laboratory discussions is that the scientific method can be applied to geographic problems in the social sciences. Hypotheses can be drawn from existing theory, tested with existing data, and conclusions drawn about both the real world and theory. Specifically the student is to deduce a descriptive model regarding the spatial variation of a current social problem.

Current problems in U. S. society are chosen for several reasons. By contrast to other topics, student interest is sparked with discussion of such problems. Everyone has an opinion on the problems and solutions. During discussions the students also realize the policy and political implications of the many theories and models given in the lectures. The particular problem studied is up to each student. As an example, after a discussion of poverty in the United States, one student may wish to examine the distribution of family median income or percent families with incomes under \$3,000 a year, for counties in California, or median incomes for the forty largest cities of a state.

After identifying the research problem, specific testable hypotheses are constructed. Considering theory and data sources, the student develops several simple, two-variable models in the form $Y = f(X)$. From the lectures and outside readings the student is also able to state the direction of the relationship, its spatial variation, and the explanatory power of the model. One of the instructional aims is readily achieved. Most students have a preconception of the census as an all embracing source of information. Disillusionment quickly develops as they realize that few of the socio-economic variables actually measure the problems under investigation.

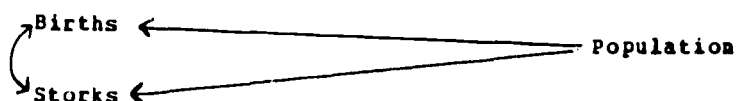
291

Association Versus Causation

The important difference between association and causation is emphasized, since few students have consciously understood it. Two examples are useful. In a region with a mountain chain, rainfall is greater at higher elevations due to cooling as clouds pass over the peaks. The association between elevation and rainfall also implies a causal chain. Increasing elevation will increase the rainfall, but increasing the rainfall of an area with artificial methods, such as a cloud seeding, will not increase the elevation of an area! In contrast, the association for villages between number of storks and number of births does not imply causation. Larger villages have a larger number of births, and a larger number of chimneys, favorite nesting places of storks, and hence larger stork populations. However, the causation is not direct; increasing stork population does not bring about more babies! Instead the "causal" chain operates through a third variable, total population. Symbolically, the simple causal chain between rainfall and elevation is $\text{Rainfall} = f(\text{Elevation})$, or if a straight line is used for causation:

Rainfall ←———— Elevation

The association, $\text{Births} = f(\text{Storks})$ actually is the causal chain:



The curved line shows association (correlation) between the two variables. The use of such diagrams greatly clarifies the construction of models and their interpretation.

Preparing the Data Deck

After designing the specific model and corresponding selection of variables by appropriate observations from the census, several mechanical steps exist. The data must be placed in eighty column coding sheets. Since less than one percent of the students have any familiarity with keypunching, copying data in tabular form precisely in appropriate columns is a very important step. The regression program, QIKREG, uses a fixed input format. The students use five variables in fields of ten columns each (5F10.0), and the last thirty columns for identifiers. They are instructed to place each data entry in the appropriate field, right or left justified (the student's choice), with a decimal point and no embedded blanks. Experiments with teaching format specifications and variable format was a disaster. A little knowledge of such things is dangerous since the students are more creative than FORTRAN is flexible. Format free input using MANELIST procedures and new options available in Release 20 level H FORTRAN also proved to be too difficult, since explanations proved too elaborate to be absorbed by the novice.

Keypunching proved easier to teach than initially expected. Shift, automatic feed, and use of the duplication key to correct errors proved to be easily learned.

To aid in correcting any keypunching errors, a special computer procedure was used to list the card deck. Essentially the program simply copies the card onto paper with the standard line printer. Such a procedure allowed greater ease in checking the deck. However, since the list procedure rarely fails, its use has an additional benefit in removing fear of the black box.

Fear and the Computer

Even in this day, despite the common use of punched cards for paying bills, and computerized registration procedures at UCLA, students still view the computer as a mystical, all-knowing being. Lectures about how a computer works does little to allay fears. Binary mathematics and switching circuits only add to the mystic. It is important to simply emphasize that the computer is only a dumb beast; it can only do, and does, exactly what it is told. For one thing, such an approach to the computer makes the student realize that when a job fails, it is his mistake and not an error in the program or the computer. The responsibility rests solely on the student to prepare his input deck carefully, correct all errors, and recheck his deck. Secondly, the approach helps to alleviate fears of the computer "knowing" everything or being capable of "thought."

Statistical Analysis

Once the basic data deck is ready for analysis, an introduction to statistics begins. Returning to the discussion of specific hypotheses, $Y = f(X)$, is restated in the linear form, $Y = a + bX$. The complete regression curve family is acknowledged, (curvilinear, polynomial, etc.),

but the simple, linear regression model with one independent variable is sufficient as an introduction.

Rather than turn to correlation, which experience showed to distract from the discussion, we continue on the simple linear model. Scatter graphs with the estimated regression line drawn demonstrates the meaning of sum of the squared deviations. Later, the same type of graphs give meaning to residuals from the derived equation. Many students are content for purposes of the laboratory discussion, that 'a' and 'b' are unique, but it helps to convince the doubters by actually showing with a little calculus how the coefficients are derived.

Once an equation is derived the statistical significance and percent variation explained are very important in model building. The significance of 'b' is explained in nonmathematical terms of confidence of 'b' as a predictor, or replicator, of the original data. A widely dispersed scattergram gives a poorer predictor 'b' than a narrow dispersal of points about the actual regression line. The QIKREG program computes the confidence limits for 'b' and prints a message for the student when the 95% level is attained or exceeded. The program also computes R-squared and prints the percent total variation explained. The relationship could be significant at the 0.05 level but still explain only 25% of the total variation. The significance and "power" of each relationship is the key to the construction of the student's spatial model. Residuals are computed for each regression as the deviations often have spatial patterns important to the interpretation and development of the model. QIKREG sample output is shown in the paper. The students prepare a brief, five-page paper reviewing the hypotheses, data sources, each regression model and results, and their model of spatial variation.

MAPPING

The second assignment is a brief report on the spatial pattern of the model. Using SYNAP the student produces contour maps of the chosen dependent variable and residuals from each of the significant relationships. SYNAP requires several input subdecks: (1) an A-OUTLINE package consisting of the keypunched arbitrary grid coordinates describing the outline of the study area to be mapped, (2) -DATA POINTS contains the coordinates of the observations, such as centroids of counties or location of cities, (3) E-VALUES contain the values of the variable to be mapped, and (4) F-MAP has the parameters of the map such as size, number of contour intervals, size of each interval, etc. The program is very flexible, but for simplicity, the students are told about only a small number of options. The visual display of variation is extremely useful to the student in drawing conclusions about his spatial model.

The computer programs

All student jobs are run on UCLA's Campus Computing Network's IBM 360 model 91 with 4,000K storage available to the user. A listing of QIKREG, written by the author, is given at the back of this paper. SYNAP was written by Harvard's Laboratory for Computer Graphics. Load modules for QIKREG and SYNAP are kept on resident disk packs and are invoked by simplified procedures which eliminate JCL problems for the student. The QIKREG program, written for 100 observations, needs only 128K of storage and the average student cost for four regression models is \$0.40 for 1 second and 100 IO requests. SYNAP by comparison is very expensive. Using 250K (with no overlays) the average student job uses about 120 seconds and 2,000 IO requests, or about \$10.00 for four maps. With an average of thirty students each quarter, the average overall cost is roughly \$300 for the quarter-course.

Success

Based on examination performance, interviews, and questionnaires, of over 120 students during the past year and one-half, the success of the spatial model building laboratory with a computer has been very great. The pace of the laboratory discussions with aid from the teaching assistant keeps even the most nonscientific student interested. At the same time the ability to experiment and explore the regression technique and the numerous options of SYNAP keep the advanced and relatively computer-sophisticated student excited. The students realize from their research that scientific methodology can be applied to social geography and that theory can be tested, refined, and developed further. The difference between association and causation becomes fully appreciated and the students gain a more critical approach to theory in the social sciences. An introduction to statistical techniques and mapping with a computer allows higher level courses to cover more advanced topics since fundamentals are more easily absorbed. Learning the limitations of the census is an aid to some students in later research. Key punching is a skill many students find useful, and for a few even profitable during their college years. At least one student even changed career plans, and now is employed as a computer programmer for a planning firm producing computer maps.


```

C      ALTHOUGH THIS PROGRAM HAS BEEN TESTED BY ITS CONTRIBUTOR, NO      QKRG0100
C      WARRANTY, EXPRESSED OR IMPLIED, IS MADE BY THE CONTRIBUTOR OR THE QKRG0120
C      UNIVERSITY OF CALIFORNIA, AS TO THE ACCURACY AND FUNCTIONING OF   QKRG0140
C      PROGRAM AND RELATED PROGRAM MATERIAL, NOR SHALL THE FACT OF THE  QKRG0160
C      DISTRIBUTION CONSTITUTE ANY SUCH WARRANTY, AND NO RESPONSIBILITY  QKRG0180
C      IS ASSUMED BY THE CONTRIBUTOR OR THE UNIVERSITY OF CALIFORNIA, IN QKRG0200
C      CONNECTION THEREWITH.                                           QKRG0220
C                                                                           QKREG
C      LANKFORD, DEPT OF GEOG. UCLA                                     QKRG0240
C      OCTOBER 1971                                                    QKRG0280
C      DIMENSION T(20),FMT(20),X(100),Y(100),YC(100),RESID(100),DT(100,7) QKRG0300
C      DIMENSION XYV(100,5)                                           QKRG0320
C      REAL*8 XNAME,YNAME                                             QKRG0340
C      DATA BLK/' ',/,'00/'?????????/'/                               QKRG0360
1000  READ (5,19,END=2000) T                                          QKRG0380
19    FORMAT (20A4)                                                  QKRG0400
      READ (5,32,END=3000) N,MXN                                       QKRG0420
32    FORMAT (11,2X,13)                                             QKRG0440
C                                                                           QKRG0460
      WRITE (6,101) T,N,MXN                                             QKRG0480
101   FORMAT ('1',20A4//'NUMBER OF VARIABLES ='13/'NUMBER OF OBSERVATI QKRG0500
      *ONS ='13)
C      ERROR SECTION
      IF (N.GT.5) GO TO 201                                           QKRG0540
      IF (N.LT.1) GO TO 202                                           QKRG0560
      IF (MXN.GT.100) GO TO 203                                       QKRG0580
      IF (MXN.LT.1) GO TO 204                                         QKRG0600
      GO TO 400                                                         QKRG0620
C                                                                           QKRG0640
201   WRITE (6,2010) N                                               QKRG0660
2010  FORMAT('NUMBER OF VARIABLES GREATER THAN ALLOWED',I6)         QKRG0680
      STOP                                                             QKRG0700
202   WRITE (6,2020) N                                               QKRG0720
2020  FORMAT('NUMBER OF VARIABLES LESS THAN ONE',I6)                 QKRG0760
      STOP                                                             QKRG0780
203   WRITE (6,2030) MXN                                             QKRG0800
2030  FORMAT('NUMBER OF OBSERVATIONS GREATER THAN ALLOWED',I6)     QKRG0820
      STOP                                                             QKRG0840
204   WRITE (6,2040) MXN                                             QKRG0860
2040  FORMAT('NUMBER OF OBSERVATIONS LESS THAN ONE',I6)             QKRG0880
      STOP                                                             QKRG0900
400   DO 60 I=1,MXN                                                  QKRG0920
60    READ (5,205) (XYV(I,J),J=1,5),(DT(I,K),K=1,7)                 QKRG0940
205   FORMAT(5F10.0,7A4)                                           QKRG0960
      DO 1500 KK=1,7                                                  QKRG0980
      SX=0.                                                            QKRG1000
      SY=0.                                                            QKRG1020
      SXY=0.                                                           QKRG1040
      SX2=0.                                                           QKRG1060
      SY2=0.                                                           QKRG1080
      SDEVX=0.                                                         QKRG1100
      SDEVY=0.                                                         QKRG1120
C                                                                           QKRG1140
C      OPERATIONS BEGIN HERE
C      PROBLEM CARD
      READ (5,20,END=2000) IDEP,INDFP,YNAME,XNAME                    QKRG1160
C                                                                           QKRG1180
20    FORMAT (211,2A8)                                              QKRG1200
      IF (XNAME.EQ.BLK) XNAME=QQ                                       QKRG1240
      IF (YNAME.EQ.BLK) YNAME=QQ                                       QKRG1260
      IF (IDEP.LT.1) GO TO 501                                         QKRG1280
      IF (IDEP.GT.N) GO TO 501                                         QKRG1300
      IF (INDEP.LT.1) GO TO 502                                       QKRG1320
      IF (INDEP.GT.N) GO TO 502                                       QKRG1340
      GO TO 550                                                         QKRG1360
501   WRITE (6,5010) IDEP                                           QKRG1380
5010  FORMAT('DEPENDENT VARIABLE INDEX INVALID FOR THIS ANALYSIS ='I6/ QKRG1400
      *GOING TO NEXT CARD')
      GO TO 1500                                                       QKRG1440

```



```

502 WRITE (6,5020) INDEP QKRG1460
5020 FORMAT('0INDEPENDENT VARIABLE INDEX INVALID FOR THIS ANALYSIS =',QKRG1480
*16/'GOING TO NEXT CARD') QKRG1500
GO TO 1500 QKRG1520
550 DO 73 J=1,MX0 QKRG1540
X(J)=XYV(J,INDEP) QKRG1560
73 Y(J)=XYV(J,INDEP) QKRG1580
WRITE (6,30) INDEP, YNAME, INDEP, XNAME QKRG1600
30 FORMAT('0', 'DEPENDENT VARIABLE NO.',19,2X,A8/1X,'INDEPENDENT VQKRG1620
*ARIABLE NO.',18,2X,A9) QKRG1640
C QKRG1660
C QKRG1680
DO 58 K=1,MX0 QKRG1700
SX=SX+X(K) QKRG1720
SY=SY+Y(K) QKRG1740
SXY=SXY+(X(K)*Y(K)) QKRG1760
SX2=SX2+(X(K)**2) QKRG1780
58 SY2=SY2+(Y(K)**2) QKRG1800
C QKRG1820
XBAR=SX/MX0 QKRG1840
YBAR=SY/MX0 QKRG1860
SMX2=SX2-XBAR*SX QKRG1880
SMY2=SY2-YBAR*SY QKRG1900
SMXY=SXY-XBAR*SY QKRG1920
VARY=SMY2/MX0 QKRG1940
VARX=SMX2/MX0 QKRG1960
SDEVY=SQRT(VARY) QKRG1980
SDEVX=SQRT(VARX) QKRG2000
R=SMXY/SMX2 QKRG2020
A=YBAR-(B*XBAR) QKRG2040
FV=R*SMXY QKRG2060
R2=FV/SMY2 QKRG2080
R=SQRT(R2) QKRG2100
R=SIGN(R,R) QKRG2120
SSDEV=0. QKRG2140
DO 59 L=1,MX0 QKRG2160
YC(L)=A+B*X(L) QKRG2180
RESID(L)=Y(L)-YC(L) QKRG2200
59 SSDEV=SSDEV+(Y(L)-YC(L))**2 QKRG2220
C QKRG2240
SFRR=SQRT(SSDEV/MX0) QKRG2260
SFRR=SFRR/(SDEVX*SQRT(FLOAT(MX0))) QKRG2280
C QKRG2300
WRITE (6,3) YBAR,XBAR,SDEVY,SDEVX,A,B, R2,R QKRG2320
3 FORMAT ('0',53(1H-),'STATISTQKRG2340
*ICAL ANALYSIS',54(1H-)//10X,'MEAN Y =',F20.5/10X,'MEAN X =',F20.5QKRG2360
*//10X,'STANDARD DEVIATION OF Y =',F20.5/10X,'STANDARD DEVIATION QKRG2380
*F X =',F20.5//10X,'Y INTERCEPT =',F14.5/10X,'B =',F24.5// QKRG2400
*//10X,'COEFFICIENT OF DETERMINATION =',F15.5/10X,'CORRELATION COEFQKRG2420
*FICIENT =',F20.5//) QKRG2440
C QKRG2460
IF (ABS(R).GE.1.98*ABS(SFRR)) WRITE (6,5) QKRG2480
5 FORMAT ('0R EXCEEDS TWICE ITS STANDARD ERROR'/1X,'RELATIONSHIP CONQKRG2500
*FIDENT AT 95% LEVEL'2X,'CONGRATULATIONS') QKRG2520
C RESIDUALS QKRG2540
C QKRG2560
WRITE (6,14) QKRG2580
14 FORMAT('0',40X,'RESIDUALS FROM REGRESSION'//32X,'OBSERVED VALUE',6QKRG2600
*X,'COMPUTED VALUE',11X,'RESIDUAL'/1X,'CASE',15X,'X',18X,'Y',15X,'QKRG2620
*Y(C)',19X,'Y-Y(C)') QKRG2640
DO 96 L=1,MX0 QKRG2660
WRITE(6,27) L,X(L),Y(L),YC(L),RESID(L),(DT(L,K),K=1,7) QKRG2680
27 FORMAT(1X,14,4F20.3,5X,'('7A4,')') QKRG2700
96 CONTINUE QKRG2720
1500 CONTINUE QKRG2740
GO TO 2000 QKRG2760
3000 WRITE(6,3010) QKRG2780
3010 FORMAT('0SOMETHING FOULED UP. END OF DECK WHILE TRYING TO READ COQKRG2800
*NTRL CARD ') QKRG2820
2000 STOP QKRG2840
END QKRG2860

```



QIKREG SAMPLE OUTPUT

DEPENDENT VARIABLE: MO. RENT
 INDEPENDENT VARIABLE: SQ. FT. INC.

----- STATISTICAL ANALYSIS -----

MEAN Y = 87.00000
 MEAN X = 7214.63672

STANDARD DEVIATION OF Y = 13.15322
 STANDARD DEVIATION OF X = 922.09180

Y INTERCEPT = 9.04353
 S = 0.01090

COEFFICIENT OF DETERMINATION = 0.98410
 CORRELATION COEFFICIENT = 0.99202

S EXCEEDS TWICE ITS STANDARD ERROR
 RELATIONSHIP COEFFICIENT AT 95% LEVEL CONGRATULATIONS

RESIDUALS FROM REGRESSION

CASE	X	OBSERVED VALUE Y	COMPUTED VALUE YCI1	RESIDUAL Y-YCI1	
1	6188.000	75.000	76.505	-1.505	ALAMEDA
2	7185.000	82.000	87.374	-5.374	ALHAMBRA
3	7675.000	95.000	92.171	2.829	ANAHUET
4	9526.000	100.000	112.095	-12.095	ARCADIA
5	6941.000	68.000	90.353	-22.353	BAGBY SPICEL
6	6126.000	82.000	76.489	5.511	BALDWIN GARD
7	6814.000	87.000	85.547	1.453	BELLFLOWER
8	6574.000	81.000	80.755	0.245	BORNELEY
9	11977.000	122.000	139.616	-17.616	BOUNTY HILLS
10	7810.000	93.000	90.917	2.083	BURBANK
11	7577.000	87.000	93.610	-6.610	BURBANK
12	6969.000	100.000	85.019	14.981	CANAL VISTA
13	6796.000	80.000	77.246	2.754	COMPTON
14	7323.000	96.000	88.478	7.522	CORONA
15	6781.000	94.000	82.969	11.031	COSTA MESA
16	7862.000	91.000	94.754	-3.754	CULVER CITY
17	8000.000	104.000	96.259	7.741	DAILY CITY
18	8262.000	112.000	99.148	12.852	DOWNEY
19	7021.000	95.000	85.886	9.114	EL CAJON
20	8159.000	90.000	97.949	-7.949	EL CERRITO
21	6606.000	75.000	81.067	-6.067	EMERYVALE
22	7141.000	88.000	86.184	1.816	EMERYVALE
23	6104.000	68.000	75.643	-7.643	FRESNO
24	7992.000	91.000	96.183	-5.183	FULLERTON
25	7741.000	94.000	93.435	0.565	GARDENA
26	7690.000	100.000	90.285	9.715	GARDEN GROVE
27	7561.000	81.000	91.495	-10.495	GLENDALE
28	7645.000	90.000	92.389	-2.389	ANTHONY
29	7204.000	97.000	87.581	9.419	INYARD
30	8285.000	75.000	77.582	-2.582	MOUNTAIN VIEW PARK
31	7104.000	92.000	91.686	0.314	INGLEWOOD
32	7923.000	91.000	95.419	-4.419	LA HABRA
33	7600.000	114.000	91.898	22.102	LA HABRA
34	7907.000	98.000	95.245	2.755	LA JOLLA
35	6170.000	75.000	80.649	-5.649	LONG BEACH
36	6876.000	78.000	84.225	-6.225	LA
37	7187.000	86.000	87.361	-1.361	LYNSWOOD
38	8789.000	109.000	99.130	9.870	LYNNHURST BEACH
39	8181.000	115.000	96.561	18.439	LYNNWOOD PARK
40	6357.000	61.000	78.347	-17.347	MONTEVISTA
41	6630.000	79.000	81.323	-2.323	MONTEVISTA
42	7351.000	84.000	89.184	-5.184	MONTEVISTA
43	7680.000	85.000	92.443	-7.443	MONTEREY PARK
44	7666.000	108.000	91.527	16.473	MT VIEW
45	5474.000	76.000	64.811	11.189	NATIONAL CITY
46	8971.000	104.000	102.484	1.516	NEWPORT BEACH
47	7015.000	95.000	85.571	9.429	IRVING
48	6303.000	73.000	77.758	-4.758	OAKLAND
49	6605.000	74.000	78.070	-4.070	ONTARIO
50	8737.000	85.000	82.440	2.560	ONTARIO
51	6671.000	76.000	78.590	-2.590	OSHAHO
52	6132.000	104.000	109.600	-5.600	PALO ALTO
53	6730.000	81.000	76.963	4.037	PARAMOUNT
54	8922.000	80.000	85.207	-5.207	PARAMOUNT
55	7065.000	88.000	86.109	1.891	PERCA DIVINA
56	6595.000	78.000	80.833	-2.833	PERRIS
57	6660.000	71.000	81.650	-10.650	PERRIS
58	8880.000	91.000	85.049	5.951	REDONDO BEACH
59	7769.000	103.000	95.523	7.477	REDONDO CITY
60	6851.000	80.000	82.723	-2.723	RICHMOND
61	6807.000	82.000	83.253	-1.253	RIVERSIDE
62	6961.000	72.000	84.736	-12.736	SACRAMENTO
63	7035.000	76.000	85.718	-9.718	SALINAS
64	6125.000	69.000	75.218	-6.218	SAN BERNARDINO
65	7663.000	101.000	91.856	9.144	SAN BRUNO
66	7389.000	77.000	85.598	-8.598	SAN CLEMENTE
67	6011.000	85.000	81.189	3.811	SAN DIEGO
68	6717.000	73.000	82.272	-9.272	SAN FRANCISCO
69	6644.000	90.000	86.801	3.199	SAN JOSE
70	7330.000	93.000	88.955	4.045	SAN LEANRO
71	8256.000	116.000	98.852	17.148	SAN MATEO
72	6104.000	81.000	77.769	3.231	SANTA ANA
73	6477.000	90.000	79.655	10.345	SANTA ANA
74	7477.000	98.000	90.503	7.497	SANTA ANA
75	8287.000	67.000	84.737	-17.737	SANTA ANA
76	6463.000	91.000	83.647	7.353	SANTA ANA
77	6562.000	76.000	80.366	-4.366	SANTA ANA
78	6897.000	79.000	84.180	-5.180	SANTA ANA
79	7578.000	85.000	91.655	-6.655	SANTA ANA
80	6059.000	55.000	75.079	-20.079	SANTA ANA
81	7937.000	107.000	95.572	11.428	SANTA ANA
82	8050.000	96.000	96.804	-0.804	SANTA ANA
83	6606.000	66.000	81.062	-15.062	SANTA ANA
84	6980.000	111.000	102.582	8.418	SANTA ANA
85	6587.000	92.000	86.656	5.344	SANTA ANA
86	7750.000	84.000	89.424	-5.424	SANTA ANA



THE COMPUTER IN UNDERGRADUATE GEOGRAPHY AT MIDDLEBURY

Vincent H. Malstrom
Middlebury College
Middlebury, Vermont 05753
Telephone: (802) 388-4051

The use of the computer in the processing and graphic presentation of geographic data is not only a well-established practice today, it is also one of rapidly growing importance. Many research projects which were scarcely conceivable in the years B. C. ("before the computer") are now commonplace, though one would hesitate to call them routine. Moreover, the availability of sophisticated programs such as SYMAP and SYMVU permit the graphic portrayal of information of interest to the geographer so readily and so effectively that whole new dimensions of cartography are being opened. Thus far, however, as is to be expected, the principal utilization of this exciting tool has been at the graduate and post-graduate levels of research and teaching. Seldom is the undergraduate exposed to a computer, except perhaps in some game-playing situation, and then usually in a fairly straight-forward question-and-answer relationship. It is the author's belief, however, that the undergraduate student should become acquainted with this powerful tool as soon as possible in his academic career and that he should do this, insofar as possible, by employing it in research projects of his own design. With these goals in mind, the author has developed several programs which have been used with considerable success in introductory as well as advanced geography courses at Middlebury College.

It should be stressed at the outset, however, that the form which undergraduate involvement with the computer has taken at Middlebury is the direct result of the facilities available to our students. Middlebury College very early joined the Dartmouth Time-Sharing System and has since enjoyed the manifold capabilities which the Kievit Computation Center affords. Among the most obvious advantages of this association are: (1) the use of BASIC as the principal language, allowing students with no previous computer experience to successfully run programs after only a minimum of instruction; (2) a direct and instantaneous student contact with the computer, obviating the need for punching decks of cards, delivering them, and calling for the results later; (when using the teletype and BASIC, the student is immediately informed of any format errors he may have made, and after the appropriate corrections have been undertaken, the running of the program yields immediate results); and (3) because the College has contracted for blocktime, the only constraint on the student-use of the computer is the availability of a teletype console - in contrast to other situations where limits to use are imposed by financial considerations. It is against this background, therefore, that the computer experience in undergraduate geography at Middlebury must be viewed.

In the introductory geography course, which averages between 180 and 220 students per year, the first computer assignment normally involves a climatic analysis. For this purpose, a classification system recently developed by the author has been employed. Indeed, the classification itself was developed with the assistance of the computer, and student research projects have largely been geared to testing new data and verifying results with the system. Knowledge of this fact has enhanced the students' interest in and appreciation of the project, which is not viewed as being simply something to do as a course requirement but as an endeavor having broader professional and practical applications. After the student has called the program (entitled MACLIM) from the Dartmouth Library, he inserts the appropriate data for his station and is informed in return of the Water Need, the Warmth and Moisture Indices of the station, and its climatic classification. (Having previously spent from 15 to 30 minutes manually working out similar results, he is, of course, immensely impressed to obtain the same data, usually more accurately calculated, in a quarter of a second!) Moreover, the computer then gives him the option of having the Water Balance of the station analyzed on a month-by-month basis, and beyond that, when the teletype is coupled to an HP-7000A plotter, of having the Water Budget graphed out for him. Using one or more of these computer-generated materials, the student is then called upon to identify the climatic station (if this information has not been previously provided for him) and to interpret the results in terms of the land-use potential of the region, and the availability of water for irrigation, power, etc. When the results of individual student projects are later drawn together and compared, broader regional differences in climate are quickly discovered and possible explanations are offered. In this manner the student comes to appreciate the computer as a valuable analytical tool, in this instance for enhancing his understanding of the variations through space of climatic phenomena.

The geographer's concern with the spatial analysis of data is effectively illustrated by yet another use of the computer in our introductory course, namely in the production of 'maps' whose content the student is called upon to interpret and explain. For this exercise, data not commonly portrayed on conventional atlas maps are stored in a series of files, ten items to a file. (For this reason alone the student is aware that he is creating an 'original' map with the help of the computer.) In one recent exercise, for example, three such files were used for a class of over 200 students, resulting in approximately seven printouts for each set of data.

Because the assignments were randomly made, there was little likelihood of collaboration between students working on the same data, and, due to the independent nature of the assignment, little consequence of such collaboration had indeed taken place.

Before the exercise itself is described more fully, a word should be said about the mapping program upon which it is based. Developed by the author in collaboration with some of his advanced students, the so-called CONMAP program makes no pretense of providing so sophisticated a product as that produced by SYNAP, for example. In its present form, however, CONMAP does provide the student with an opportunity to prepare his own maps on a teletype printer, with all the limitations inherent in such a piece of hardware, including narrow map format, relatively slow printout, and a restricted choice of printing tones**. Among the options open to the student with the CONMAP program are (1) the spatial ordering of data, either in 'raw' or manipulated form (the so-called "data-point" map); (2) the production of a "symbols" map, on which squares proportional to the data are drawn by a plotter; and (3) the production of a "choropleth" map on which areas are shaded according to the classes of data which each of them represents. If the user specifies the latter option, he has the further choice of having the data analyzed by quartiles, quintiles, or by classes whose intervals he may wish to specify himself, such as 30%, 40%, etc. For the first two kinds of maps, two files are used, the so-called A-File, which provides the names and coordinates of all the places to be shown on the map, and the B-File, which contains up to ten items of data to be mapped for each place. In the preparation of a choropleth map, a third, or C-File, is required to outline the areas to be shaded.

Whether the student elects to produce a data-point, symbol, or choropleth map will depend on the use he intends to make of it. While the symbol and choropleth maps both have greater visual impact than the data-point map, the latter can be easily contoured and likewise permits further statistical analysis. Moreover, it is the fastest to print out, and hence has been the type of map most frequently used in our beginning classes. (To cite a specific example, a recent exercise which mapped data for the 58 counties in the State of California averaged about 1/2 second of computer time and took about 1 and 1/2 minutes to print out.) On the other hand, as upper-classmen working on a term paper in a regional geography course used the "symbols" option to produce an urban atlas of the entire Soviet Union. In it he not only depicted every city over 100,000 population for each census from 1897 on, but by a simple manipulation of the data he also showed their absolute growth during each intercensal period. Thus, by using the CONMAP program, he accomplished in a few days' time what would have taken literally months to do by hand--and again, with far greater accuracy than if it had been carried out manually by a cartographer. Thanks to the computer, he had produced a valuable piece of original research, which, when suitably edited and re-drafted, could conceivably form the basis of a professional paper of some merit.

Getting back to the computer mapping exercise in the large introductory course, once the student had prepared his data-point map, his assignment was to explain the patterns he found using any other source materials which he could uncover. Many of them looked for clues in the printouts of computer maps dealing with other, related subjects, as well as in textual and cartographic materials available in the library. Almost without exception, the students experienced a real 'involvement' with their respective projects, bringing to bear in their analyses virtually all the principles that had been discussed through the term. Indeed, many of them turned in highly perceptive studies, accompanied by finished maps which were based on the computer data.

- * Although the challenges confronting the 'teletype cartographer' are somewhat akin to those which the Romans faced in using rolls of papyrus, maps of any width can be produced by pasting the strips together.
- ** Although over-printing is possible for emphasizing tonal differences, its use greatly slows down an already slow printout.

In the illustrations cited above, the use of the computer has provided the undergraduate student with critical assistance in solving problems within the context of our regular geography curriculum. However, a far more ambitious computer-oriented course was developed and offered by the author during the Winter Term of 1972. (During this five-week period, the student concentrates on a single course of study.) Entitled "World 2000," this course was limited to an enrollment of 20. By drawing lots, each student was assigned to represent one of the twenty most populous nations in the world. Although the size of the country's population was the chief criterion for including it in the list, it was felt that they would provide a wide range of diversity in levels of economic development as well, ranging from the United States and the Soviet Union at one extreme to China and India on the other. After completing a 'national inventory' for each of the countries by collecting statistical data on virtually all aspects of its economic, social, and political life, the students utilized the computer to manipulate these data for purposes of projection and analysis. For example, using the climate classification described earlier as a basis, they ran a program called WEATHER to determine whether their crop yields would be up or down from one year to the next. Inasmuch as each class day was intended to represent a one-year projection into the future, when the students came to class each day, the weather summary for the preceding 'year' for a representative station in their country was ready for their analysis*. Having already run another program titled FORCAST to determine what his grain (and other commodity) needs would be, both at existing (i.e. 1971) and improved levels of consumption--the student would quickly ascertain whether his harvest would be adequate to meet his own requirements or by how great a margin it fell short of or exceeded these needs, thereby necessitating imports or permitting exports. This in turn would engage his country in interaction with other countries through the media of trade, for which another program (by that name) was available for 'bookkeeping' purposes. Another program entitled BUDGET established the financial resources available to each country and provided each player with the option to budget them as he saw fit. Yet other programs provided a quantification of the effectiveness of given policies instituted by the different governments and an annual overview of the internal and external political situations. Likewise, each "year" the players ran a program entitled SUMMARY to take stock of their government's economic and political progress during the preceding 'twelve months.' Positive achievements were registered in increased GNP's and per capita incomes, and in the increased support of the people, whereas stagnation or regression was reflected in an eroding economy and political disaffection. As the plight of some countries became increasingly desperate with time, they called for the option of using force in their dealings with one another, and asked that a WAR program be created a request which necessarily was met but with some hesitance and regret.

Naturally, the outcome of such a course is strongly dictated by the personalities and decisions of the individual players, as indeed, is the course of human events in the real world. No doubt very different results would have been forthcoming had twenty other students played each of the respective roles, or even if the same twenty students had each represented a different country. Nevertheless, at the termination of the course, all were agreed that it had provided an incomparable learning experience, not only giving them a far greater understanding of the major problems confronting the nations of the world today, but also a much deeper appreciation of how individual, momentary decisions materially affect long-term results. Everyone likewise came away from the course cognizant of the central role the computer had played in it, not only as a 'purveyor of facts' on which to base decisions, but also as a dispassionate 'judge' of such varied conditions as the quality of the environment in an industrialized nation to the severity of malnutrition in an underdeveloped country. Indeed, it should be emphasized that the design and execution of this winter term course would have been quite impossible without the computer, with which a teletype link was kept open for the entire 2 1/2 hour class period each day in order to expedite decisions and interactions. Thus, from having successfully involved students in using the computer for individual research projects as outside assignments in courses regularly offered in the fall and spring terms, the author feels that he has made a large and rewarding advance toward structuring an entire course around its use during the shorter winter term. On the basis of the favorable response of the students, and with an increased measure of "wisdom" born of this experience, he confidently anticipates the expanded use of a "computer-oriented approach" in his regular course offerings in the future.

* Here, actual data was used from a series of years in the past. Obviously, no student was allowed "advanced warning" of any oscillation in the climate, but was obliged to adapt 'after the fact.'

THE USE OF COMPUTERS IN GEOGRAPHIC INSTRUCTION
 AS A MEANS FOR
 STIMULATING INTEREST IN STATISTICAL METHODS

David J. Cowen and Paul E. Lovingood, Jr.
 University of South Carolina
 Columbia, South Carolina 29206
 Telephone: (803) 777-5234

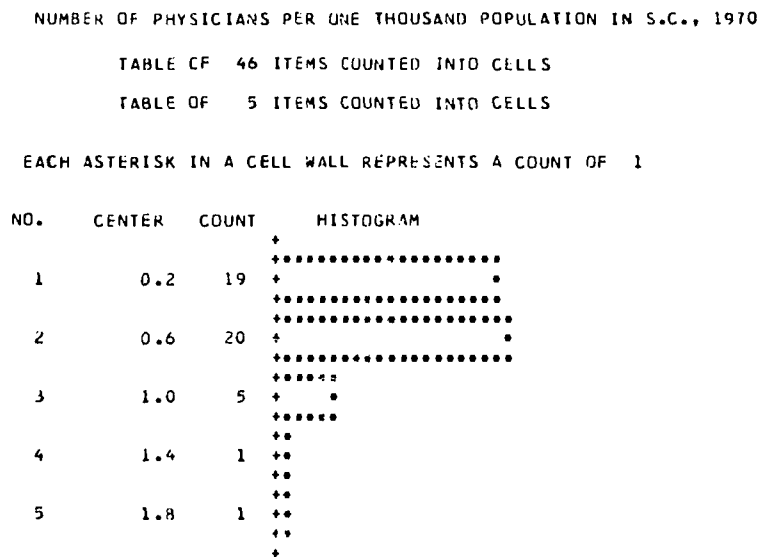
A major problem confronting anyone teaching a required undergraduate statistics course to students in the social sciences is overcoming the apathy concerning the subject matter. In teaching such a course to undergraduate geography majors for the past two years, we have found that by having the students actually collect their own set of data and analyze it with the aid of the computer, interest has been stimulated and the student has attained a sense of accomplishment. The purpose of this paper is to discuss some of the specific ways we have utilized the computer in overcoming the apathy of the students to statistics.

In order to create interest and a sense of self-confidence each student in the first week of class is required to collect a set of his own data. Because this is a course in geography the observation units must be geographical areas such as counties, cities, or states. For each of the observations five or six variables are assembled into a data matrix. It should be noted that the students are encouraged to collect data for the local region or the state. The student is therefore familiar with the data and interested in the findings. The student is instructed in data coding and keypunch operation. After punching his data he runs simple canned reproduction and data description programs. We have found that the student reacts well to this initial assignment and enjoys getting his first computer printouts. It also is helpful when covering measures of central tendency and variance for the student to have his own printouts with him in class. In addition, the printout provides a check for him when working out computations on a desk calculator.

This year in conjunction with the statistics course we ran a non-credit FORTRAN programming course for students in the department. We found that in five or six sessions the student could learn to write his own program for computing some of the simple descriptive statistics. The students who diligently participated in the programming course developed a better comprehension of the statistical methods. It was especially useful to relate programming statements to the corresponding mathematical procedures.

In discussions of data displays we have utilized a canned histogram program as demonstrated in Figure 1. The printout is utilized as an aid to the student in preparing his own displays.

FIGURE 1.



PREPARED BY MICHAEL TREADWELL FOR GEOGRAPHY 531, FALL 1971
 UNIVERSITY OF SOUTH CAROLINA

In geography point distributions are significant. Therefore, we spend a few lectures dealing with quantitative analysis of these distributions on maps. At this time the student collects another set of data pertaining to the spatial distribution of some phenomenon such as playgrounds, grocery stores, or banks. He is then introduced to cartesian coordinates and runs a program that generates nearest neighbor and bivariate descriptive statistics. Based on this output the student writes a short paper in which he discusses the distribution of the phenomenon and interprets the implications of his findings.

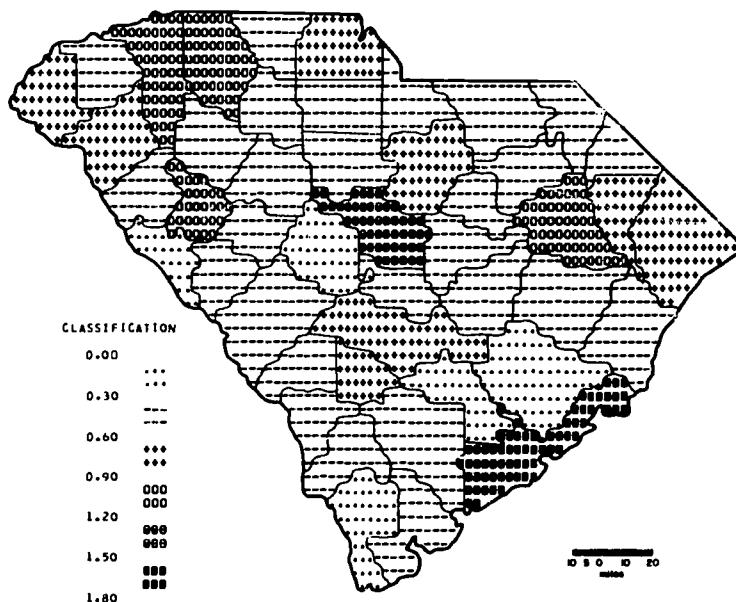
The last part of the course is devoted to analysis of variance and simple linear regression models. The student develops his own hypotheses pertaining to regional differences between areas in his study region. For example, a certain group or region of counties might be expected to differ significantly from another with respect to certain socio-economic variables. In the case of South Carolina many of these regional differences pertain to urban versus rural or piedmont versus coastal plain counties. The hypothesized regional differences are examined by the student with the aid of a canned analysis of variance program. As in all exercises the student is introduced to the method in class and works out his own results on the desk calculator to gain a better understanding. The computer output serves as a check on his own calculations.

The discussion of regression usually proves to be of most interest to the student. He usually has assembled his initial data set with the idea of examining the relationship between two socio-economic variables such as racial mix and income level. Some students originally gather their set of data rapidly and without much thought and therefore decide to collect more meaningful data for this section. Although the theoretical structure of statistical hypothesis testing is stressed in class, extensive development of the hypothesized relationship is not deemed necessary for this exercise. The output from a regression program serves as an important teaching aid in the discussion of estimating parameters for the least squares regression line and variance explanation.

As a final assignment the students map two of the variables and examine the pattern of residuals from the regression line. We have developed several computer mapping routines for the local area to aid in this final project. To undergraduate geography students producing and seeing their own data mapped by a computer proves to be quite exciting. Figures 2, 3, and 4 are examples of such maps produced by a student who is interested in the relationship between the distribution of physicians and urbanization within the state of South Carolina. Some of the students who worked through the FORTRAN course have been developing their own base maps. However, we have found it more efficient to let most students use our existing programs.

FIGURE 2.

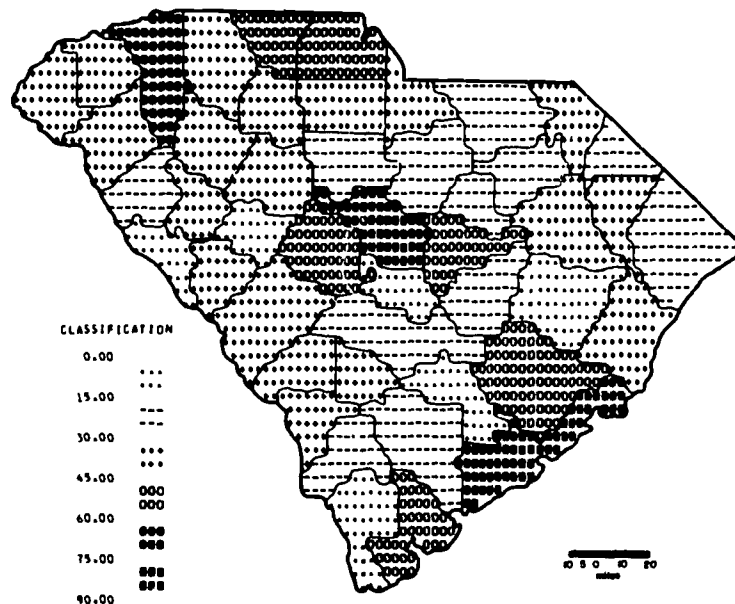
NUMBER OF PHYSICIANS PER ONE THOUSAND POPULATION IN S.C., 1970



PREPARED BY MICHAEL TREADWELL FOR GEOGRAPHY 531, FALL 1971
UNIVERSITY OF SOUTH CAROLINA

FIGURE 3.

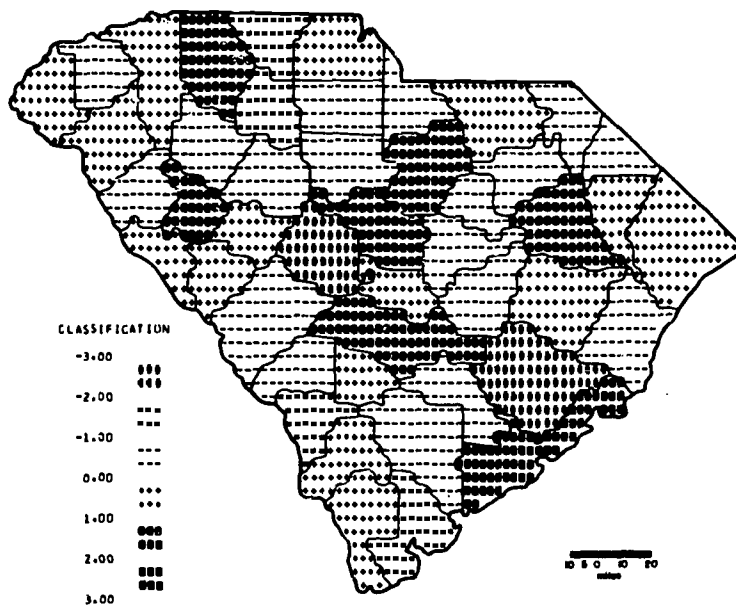
URBAN POPULATION AS A PERCENT OF TOTAL POPULATION, BY COUNTIES, IN S.C. - 1970



PREPARED BY MICHAEL TREADWELL FOR GEOGRAPHY 531, FALL 1971
UNIVERSITY OF SOUTH CAROLINA

FIGURE 4.

PERCENT URBAN POPULATION ON NUMBER OF PHYSICIANS/1000 POPULATION
STANDARDIZED RESIDUALS FROM REGRESSION, BY COUNTY - S.C. 1970



PREPARED BY MICHAEL TREADWELL FOR GEOGRAPHY 531, FALL 1971
UNIVERSITY OF SOUTH CAROLINA

At the conclusion of this course the student has actually completed a small piece of his own research. Instead of working out a set of prearranged exercises from data supplied to him he has collected and analyzed his own. By introducing them to the computer and its capabilities at the outset of the course the students gain a sense of adventure and apathy towards statistics is reduced. In addition, the computer output is essential in this type of course as an instructional tool and as a means for students to check their own exercises.

INTRODUCING UNDERGRADUATE GEOGRAPHERS TO QUANTITATIVE
ANALYSIS THROUGH A REGIONALIZATION FRAMEWORK

Nancy B. Hultquist
University of Iowa
Iowa City, Iowa 52240
Telephone: (319) 353-3131

Introduction

Analysis of spatially recorded data by means of statistical and mathematical techniques has become firmly established in geographic research with the computer as a necessary tool. Future professional geographers need to be prepared, therefore, to handle quantitative methods. For this reason, courses in quantitative geography appear in the curricula of many graduate programs in a variety of forms varying from straightforward descriptive measures through inferential statistics to advanced multivariate analysis, and most students who continue into graduate work are required to master quantitative techniques. Undergraduate teachers, however, have been slow to introduce such techniques, perhaps mainly because of their lack of technical training. Consequently, many undergraduates are not introduced either to quantitative methodology or to computer techniques. For many the adjustment to quantitative analysis at the graduate level is often traumatic. Therefore, it is proposed here that a survey of techniques of quantitative analysis and the tools for manipulating and displaying data should be introduced early in the student's career.

The objective of this paper is to examine an experimental course which was offered in the Department of Geography at Georgia State University in 1970 solely for undergraduates, not necessarily majors. The course, "Techniques of Spatial Quantification", introduced selected statistical techniques most often encountered in the geographic literature and encouraged computer usage and interpretation of results. It was offered during the fall and spring quarters. Purposefully, the course did not dwell on mathematical particulars of each technique introduced but presented an intuitive explanation related primarily to application. One objective of the course was to remove fear of statistics and computers, and at the same time to demonstrate their value as tools in spatially-oriented research. While it is possible to teach a basic statistics course without using the computer, a course of the type described here must concentrate on computer involvement simply because of the scope. It is recommended that such a course should be incorporated into all undergraduate departments, limited to majors if necessary to control enrollment.

Data Provided for Each Student

The members of the class were divided into four groups, each of which dealt with one particular region consisting of forty Georgia counties contiguously grouped from north to south (see Figure 1). Because there are 159 counties, necessarily there was one overlapping county (Gwinnett). The data for each county consisted of 45 census variables, 20 of which were percentage data used for most analyses (see Table 1). Each student had his own data deck. This provided the basis for problem exercises and for class discussion of techniques. The regions were relatively familiar to the students and were therefore conducive to a meaningful analysis of the data and of the statistical techniques introduced.

Format of Class

For the first four weeks, the class consisted of five lectures per week. During this time an historical development of the use of quantitative methods in geography, of mathematical and statistical terminology, and of basic statistics were presented. Thereafter, the course assumed a seminar-laboratory orientation. Program runs were distributed. New techniques were discussed, problems ironed out, or interesting findings discussed.

The course instruction assumed no knowledge of the computer and, in most cases, the students had no prior experience with programming. The introduction involved basic computer terminology, concepts, and use of peripheral equipment. While programming per se was not taught, certain FORTRAN specifics were introduced, including format statements, variable mode, and columnar specifications as would be encountered in making control cards for a library program. Several short programs were written in class by the instructor to indicate the type of manipulations possible in FORTRAN. The students were encouraged to keypunch one of these and run it with hypothetical data.

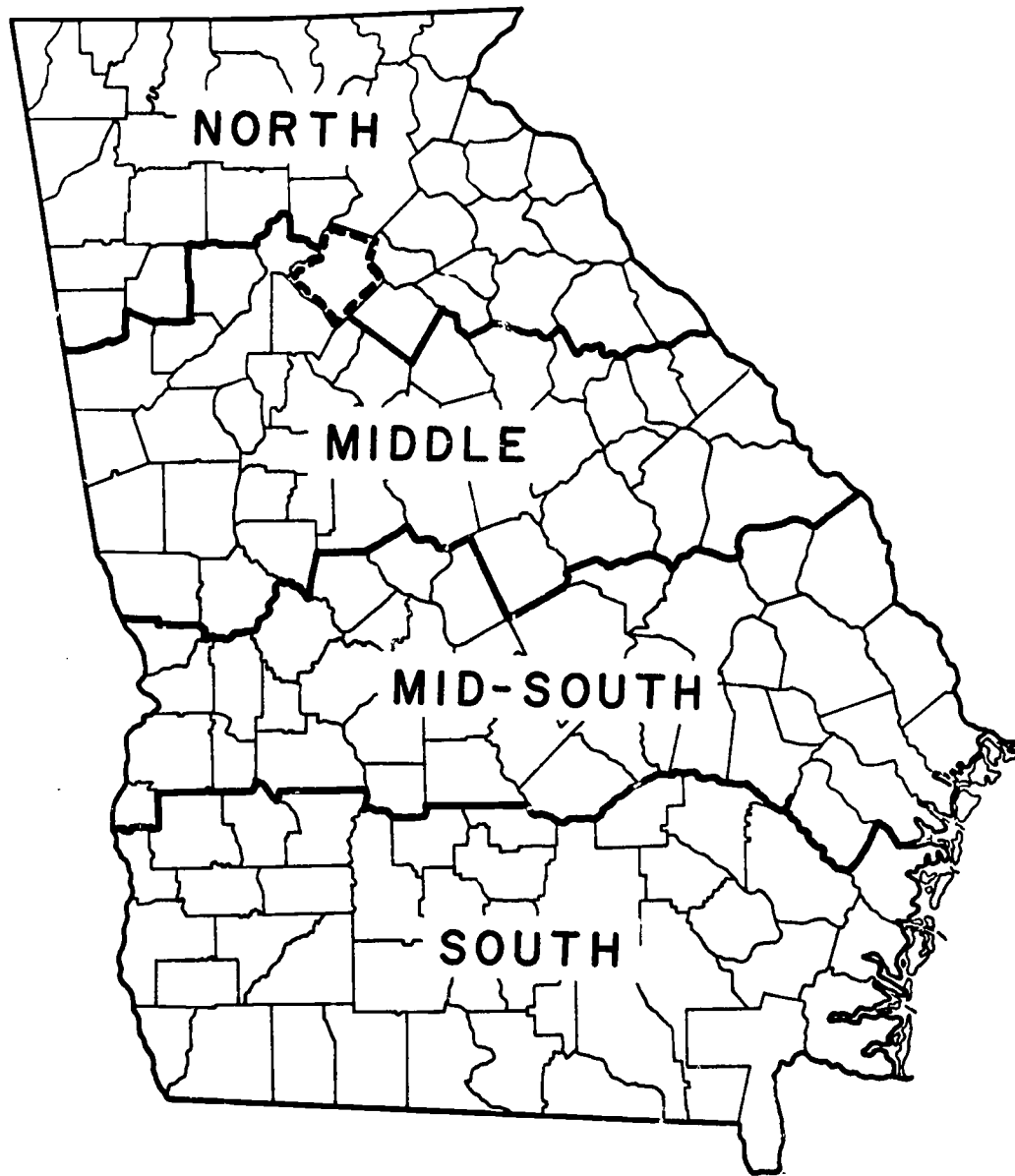


Figure 1: Four Arbitrary Study Areas Within Georgia

TABLE 1
LIST OF VARIABLES

1. population per square mile
2. percentage population change 1950-60
3. percent urban 1960
4. percent Black 1960
5. ratio of births to deaths
6. median school years completed for persons over 25 in 1960
7. per capita income
8. median family income in 1959 of families in 1960
9. percent families earning under \$3,000
10. percent families earning over \$10,000
11. percent all land in farms (1959)
12. percent all commercial farms with sales over \$10,000
13. total persons in farm operated households
14. total employment
15. percent employment in agriculture
16. percent employment in forestry and fishing
17. percent employment in construction
18. percent employment in manufacturing
19. percent employment in wholesale and retail trade
20. percent employment in services

Other programs used were arranged so that multiple data decks could be submitted simultaneously on one computer run producing an individually identified print-out for each student. The laboratory sessions were used to examine the results and to discuss regional differences displayed by the different print-outs. Students were held responsible for furnishing control cards for each program, usually consisting of an identification card, parameter card(s), and a variable format card.

Over a three-month period, many tests, techniques, and grouping procedures were examined by the students, including measures of central tendency and dispersion, normality and transformations, correlation and regression analysis, principal components analysis, and discriminant analysis.

Students were required to run their entire data deck through each program introduced, and were encouraged to make special runs to follow up hypotheses by deleting observations or variables. Since the master program read in each student's deck as data, often one student's

error would ruin the day's run for his classmates. This happened only a few times as the neophytes gained more respect for careful keypunching and verification.

Canned Programs Utilized

The first program run was a simple FORTRAN program producing means, standard deviations, standard scores, a correlation table, and reciprocal pairs of highest correlations. More time was spent on analyzing this program than any other throughout the course, because many basics were involved. Students chose variables of interest and mapped their standard scores. This provided a feeling for the counties in their study area, for example, rich vs. poor, urban vs. rural, population growth counties, educational levels, and employment patterns. Students were encouraged to do some supplemental research to substantiate quantitative findings.

After discussing the variables in relation to the counties, students examined the relationships existing among the variables. Positive and negative correlations with each variable were listed. An important subset of this list, a listing of variables with highest correlations and reciprocal pairs, was provided by the computer program. From this output a linkage analysis was performed [1]. Students were able to note the important linkages among variables over all counties, thus giving an even better picture of the characteristics in each area. At this point students had no way of mapping these findings precisely, but at least they were able to generalize linkages and to form some hypotheses which could be verified with later techniques.

The next program introduced involved an hierarchical grouping routine written by Donald Veldman[2]. This program standardizes data matrices and groups observations (or variables) using generalized distance functions to minimize total within-group variance at each step in the process. Students can map groups at various stages of the process and formulate hypotheses about the reasons for these groupings. Some students found it interesting to use variables which combined on one of the major links noted in the previous program. The hierarchical grouping program is helpful in pointing out counties which are "similar" with respect to the variables, and in allowing experimentation with variables to see how groupings change.

A multiple discriminant analysis program was introduced which allowed students to check out some of their hypotheses about appropriate homogeneous groupings based on any selected variables. The output indicated the statistical validity of their choices. Students were often temporarily shocked that the computer caught something they had overlooked, but upon reflection, they simply became more avid computer users. That was not always the case, however. Some students preferred the subjectivity they could impart to the situation.

Principal components analysis with varimax rotation was introduced next. Not surprisingly, the factors usually indicated dimensions similar to the linkage analysis results, except for a bi-polar factor which absorbed two linkages. The concepts for factor analytic interpretations were easily accepted because of the former work with linkage analysis.

Various graphic techniques were used to display the computer results, but because of lack of computer equipment, these were done by hand. More sophisticated equipment such as line plotters, cathode-ray tubes, or software such as SYMAP or MAPIT could be used with larger installations. At the time of this course, the available computer was an IBM 7040 with no plotter and with limited terminals. Students determined the high positive and negative factor loadings (.500 and above) and then named the factors. They graphed factors against one another and labeled counties, e.g. rural vs. urban. Factor scores were presented by the program. They were graphed, mapped, and interpreted further.

Results of the Course

One measure of course success came in the students' final report on activities taken to quantitatively describe their areas of Georgia. The reports indicate that a great amount of interest and work took place throughout the quarter. In addition to individual reports, each group of students also gave a resume of their findings to the entire class. This gave everyone the chance to see Georgia altogether for a change.

Another successful note was the interest generated. Persons from the fall quarter enrolled in an urban geography course the following quarter and chose topics which utilized computer techniques. There was also enough interest generated that during the spring quarter the course was offered again, and an additional advanced course was taught for some of the original members of the fall group. In the advanced class each person learned more FORTRAN, tackled a research topic of particular interest to him, and learned additional techniques.

Conclusions

Apparently, the addition of computers to the quantitative geography course offering was useful for at least two reasons. More importantly, it introduced the student to current techniques of research in the field and to terminology with which he needed to become familiar. Secondly, the personal use of the computer gave undergraduates an interest in the course and provided a necessity for keeping up-to-date on assignments. In this case computer time and money were not problems. With current financial limitations in some universities, the same course might still be taught with group rather than individual involvement. There is no doubt, however, that the individual responsibility for computer runs is a preferable situation and will produce favorable results.

REFERENCES

1. McQuitty, Louis L. (1957). "Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies," Education and Psychological Measurement, Vol. 17, pp. 207-229.
2. Veldman, Donald, Fortran Programming for the Behavioral Science, (1967), New York: Holt, Rinehart, and Winston, pp. 308-317.

CAI AND ENGLISH: A TENTATIVE RELATIONSHIP

Anna Marie Thames
Golden West College
Huntington Beach, California 92647
Telephone: (714) 892-7711

Computer programs written for remedial English students at Golden West College are designed for a specific course, English B. It is prerequisite for the college transfer English course and is required for those students who fail the college entrance test and a paragraph writing challenge. The students meet this class only twice a week for discussion and paragraph writing. In my classes the students are expected to spend at least one hour per week, for nine weeks, working on computer-programmed exercises in spelling, diction, sentence structure and paragraph construction.

The student who takes this course does so because he failed to pass an entrance examination which would have qualified him for the transfer composition course, and who subsequently enrolled for and failed English A, a grammar review. Subsequent to passing English A, he then failed to pass a challenge exam consisting of one paragraph he was asked to write. In passing this exam he could have bypassed English B. Therefore, he has failed at least twice at the college level to reach the goal of writing an effective paragraph.

Very often such a student does not have a high school diploma or has failed English in high school as well. After failing the second college challenge exam, he is scheduled to attend this class for nine weeks; his primary goal is to write a paragraph of approximately seventy-five to a hundred words which is unified, logical, and substantial.

From the beginning, the course seemed difficult to teach because, in spite of my assumptions about these students, their abilities varied greatly. They had one inability in common, however; they could not write a paragraph. After two semesters of frustrating experiences, I decided to develop other means of teaching the course, using the computer to meet individual problems, since the college did not provide tutors. In April, 1969, I asked one of my former students (who was fortunately a much better programmer than an English student) if he would teach me how to write programs for remedial English. Research suggested that little was known about the capability of the computer for use in teaching English: so the first step was to write an experimental exercise in APL. In the summer of 1969, on a Project SAL (Systems Approach to Learning) grant, we began to develop programs which would stress the principles of paragraph writing, such as unity and coherence, through drill and practice as well as programs which focused on special problems in diction. We worked beyond the summer project until November, designing about 20 programs, written for students who were ready to begin the nine-week course. At the end of that semester about half the programs were dropped, either because they were poorly designed or incompatible with the restrictions of the computer. For the last two years, the remaining programs (about 10), as well as a few new ones, have been continually revised--a critical advantage, incidentally, of CAI over programmed texts. The rationale for using the computer was based on these assumptions:

1. Students can be motivated if they are active rather than passive participants in the learning process.
2. Using a computer is a positive rather than a negative experience.
3. The text of the programs and responses should be entertaining as well as informative.
4. A few elementary programs used by students would point a direction for more sophisticated and ambitious programs later.
5. Manipulation of writing segments can make a student aware of his own weaknesses in writing.

The results were these general goals:

1. To individualize specific segments of writing skills instruction.
2. To reach the student at his level of performance.
3. To provide an immediate and re-enforcing kind of response to his efforts.
4. To shift the responsibility of learning from teacher to student.

5. To move mechanical aspects of the course out of the classroom to make room for discussion of style, viewpoint, and writing.
6. To remove the pressure of grades (by requiring only that students complete programs).

The descriptions which follow include only those programs which have been tested throughout the past three semesters by approximately 600 students.

CONFUSED is a program which asks students to distinguish between often-confused words; e.g., then-than, who's-whose, its-it's. The student is asked to type in a blank one of two words which is appropriate to the sentence. If he is correct, no explanation is given; he is presented with the next sentence. For incorrect responses, he is given a definition of each word, ("affect" and "effect," for example) and then directed to one additional sentence requiring the same words. Only ten sentences are presented if he answers them correctly, thirty if he answers them all incorrectly.

SPELLING is the name of a program which types a sentence and then asks the student to find within it a misspelled word. In the first segment of this program, the correct answers are calculated, and the words which he missed are typed out. Then he is directed to another program which includes rules for spelling certain categories of words, and requires practice in using them.

TOPSEN is a program which includes three different exercises. The first, called SPECIFY, presents a statement which the student is asked to label "S" (specific) or "G" (generalization). For each wrong answer, he is given an explanation. No score is tallied. In the second exercise, SCRAMBLE, he is presented with a paragraph whose sentences are listed in the wrong order. He is asked to identify the topic sentence of the paragraph. In the third exercise, SELECT, three sentences are typed out and the student is asked to select the one which could best be developed into a single paragraph. Neither of the last two programs include individual responses to wrong answers.

UNITY is, for the alert student, a rather short exercise since he need only answer the first two questions correctly to complete the program. He is presented with a paragraph, including one or more sentences which do not support the topic sentence. He is asked to identify them. This is an elaborately designed program in the sense that all possible responses are anticipated. If either of his first two answers is incorrect, he is presented with shorter less complex paragraphs.

PABACO is a program which operates on the same design - the student is presented with a scrambled paragraph and he is asked to reorder it so that each sentence leads logically to the next. Few students are successful with the most difficult paragraphs in this exercise. (A sample student printout is included in the appendix.)

LINKS includes two exercises. The first asks the student to find transitions included in the paragraphs presented. His answers are scored, but individual branches are not included for all possible wrong answers. The second program presents a paragraph with blanks which are to be filled in from a library of pronouns. For each wrong answer, a rule is presented explaining the reason for the correct pronoun. The second program also presents sentences with blanks to be filled in from a library of transitions.

SUBCO is one of the few manipulative programs. The student is given a complex sentence, for example, and asked to revise it as a compound sentence, using specific editing commands. Several months went into the development of these commands so that they would be written in the language of the student and would enable the text to be manipulated rapidly. If, for example, the student is asked to change the compound sentence: "He doesn't like her, but I do," to a complex sentence, he can respond with the following commands:

INSERT EVEN THOUGH BEFORE HE

DROP BUT,

the computer would then type back:

EVEN THOUGH HE DOESN'T LIKE HER, I DO.

This is a rather complicated program for the student, since he must use the right commands, manipulate phrases, and spell correctly. It was more complicated for the programmer, since he had to account in the program for mechanical errors as well as a variety of correct choices the student might make. (See appendix for samples of student copy for this program.)

In MODIFIERS, the student is again manipulating text and directing the computer. Because of his experience in SUBCO and the interest level of text here, student response is fairly positive. (See appendix for sample of student copy.)

In MISMOD, the student is asked to identify misplaced modifiers and move them. For example, in the sentence

BEER SHOULD NOT BE SOLD TO STUDENTS CONTAINING MORE THAN 3.2% ALCOHOL,

the student is first asked to identify the modifier, typing its first and last word,

CONTAINING ALCOHOL

If he does not identify this correctly, the computer will give it to him. In any case, he is next asked to re-locate it in the sentence using a caret (^) as an insert mark, in this fashion

BEER^

Wherever he places the caret, the computer will there insert the modifier already identified:

BEER CONTAINING MORE THAN 3.2% ALCOHOL SHOULD NOT BE SOLD TO STUDENTS

At this point, his revision is evaluated, and if he is incorrect, he is asked to try again. This continues for a total of three sentences.

WORDYMOD prints sentences which are obviously wordy, and the student is asked to delete all unnecessary words. He must use editing commands to accomplish this. For example, in the phrase,

TWO CHARACTERISTICS WHICH ARE NECESSARY FOR GOOD SALESMANSHIP,

the student would type

DROP WHICH ARE

and then whatever other changes he wished to make in the rest of the sentence. The computer would type back whatever changes he made, right or wrong, as soon as he typed DONE. However, the program does not allow for partial answers, and this is sometimes frustrating. By the time you read this, he may be getting credit for one change when several are programmed.

PREEMOD is a different workspace from the other two programs, so the student must load it himself. This may be the program to delete if you are pressed for time. It involves a list of sentences (^) which the student is asked to complete with one or two free modifiers. For example, he is given the sentence:

IT MAY BE SAID THAT POLICEMEN ARE RATHER SPECIAL, ALMOST ALIEN CREATURES

He is then asked to add a free modifier, in this case

WITHIN THE AMERICAN SCENE

which he can insert, using a caret

CREATURES^

Obviously there is more than one place in the sentence for this modifier, so there are no absolutely right answers. But the program disallows all answers than those designed. You may want to add more. Only two placements are allowed for this sentence,

CREATURES^

^IT

This program requires about 30 minutes' effort.

REVISE is the most sophisticated program in this group and will soon undergo major revision. The student is asked to write a paragraph constructed in this specific form: a topic sentence, three or more major supporting sentences, and three or more minor supporting sentences. Actually, he can skip any of the supporting sentences by pressing the return key on the terminal. Once he has finished, the computer will type the sentences in outline form and ask for additions or deletions. At this point, the student can make word changes easily through commands such as "replace" and "delete." Once he has completed his changes, the paragraph is typed out in revised form.

How the Programs Are Used

The programs are an integral part of the course. Each one is the subject of a week's discussion in class. A handout is prepared which introduces a particular writing principle, such as coherence, followed by various examples illustrating its application. At the conclusion of the handout, the computer program is introduced, its objectives are presented, and the student is told how to sign on to it. The programs are occasionally used as individual answers to specific problems of students in other courses.

Several programs (not listed here) focus specifically on grammatical problems not discussed in class, such as sentence fragments and the over-use of the passive voice. For example, one program asks the student to type a paragraph without any form of the verb "to be." While the program is very elementary in design, it does serve to prod the student who has this problem, but may be unaware of it.

Student Reaction

Student reaction spans the distance between extremely negative and obviously flattering responses. Among those who offer criticism, the brighter students are generally skeptical, sometimes reactionary, but frequently accurate in their comments. For example, they point out that some of the programs are written to accommodate the computer system rather than the student. Those who are less alert often complain about the complexity of the programs and the precision required to complete them successfully. These students have difficulty understanding the directions. Within this group, some of the serious students seem to value the effort made to reach them at their own ability level and the de-emphasis of competition. Within both groups, there are some who enjoy the opportunity to work independently. Most complain that the course is more difficult because of the computer programs--they cannot sit passively, as in the classroom, responding only when they wish.

Nevertheless, the students complete the programs, a requirement for passing the course, and in so doing are forced to become involved actively. Those who stay in the course improve, both in their ability to read and follow directions precisely, and to write an organized paragraph concisely. And though they have failed college work in English at least twice (these students frequently have a history of failure outside English courses as well), many are encouraged by the discovery that they are capable of interacting with a computer.

Several student paragraphs are included here, unedited, because they represent the range as well as the degree of criticism encountered thus far.

Roland Clark: I am sure the computer program I enjoyed the most was the first program I used. To be sure, having a typewriter talk back to me, and answer my questions was very amazing. One thing I like about the computers is when I answer a question wrong I am not embarrassed to death. Oh I get embarrassed all right, but I know the computer doesn't mind so it's sort of personal. All I do is try to get the computer programs, even if some take more time than they're worth.

Roger T. Brown: The computer is the unreachable instructor that will not take any backtalk from its students. If anything is typed that is not asked for, such as a smart comeback to a question, the computer just writes--incorrect command. There you sit with the computer having the last word. It only wants to hear what it has asked for and nothing else will do. I was once told that man is the only thing on this planet that can reason. After using his God given reasoning ability, he turns around and asks a computer if he is right.

Tom Beason: I chose to evaluate a computer on the basis of its relationship to man. After all the work, money, and training of personnel have gone into it, this "brain" comes out an extremely useful tool to man; always logical, never in error (if in good mechanical repair), and able to give nearly instantaneous answers. But, as a thinker, the computer is fast, efficient, and stupid; man is slow, clumsy and brilliant.

Cynthia Bakkelo: Learning with the computer is sometimes frustrating; yet it is a whole new way of learning with individual attention. It takes time to adjust to the machine; at first it was really a challenge to try and get the program started; then when it did start sometimes it would stop in the middle of the lesson and refuse to continue. Having wasted that time it was more discouraging to start all over again. The machine is not perfect nor am I; but when the problems are solved it is exciting to communicate with a machine. The machine is ideal in that it applies to individual needs; if I know the subject matter well it takes out little time to finish. The mistakes I make are carefully explained to me

and I am quizzed to see if I have understood. Although it doesn't replace the teacher it is helpful in doing the unpleasant tasks of the teacher.

During the past 18 months two surveys of student reaction have been completed. The first one, begun in spring 1971, included my personal observation log of student interaction with the computer, a collection of student paragraphs evaluating the computer, and a questionnaire which is included in the appendix. None of the students were in my classes. The primary focus of analysis on the questionnaire was on noticeably large percentages and apparent patterns of response. Initially, student answers were compiled and processed according to the percentage response on each question. A further breakdown was printed out on the basis of the grade the student expected in the course. Finally, two charts were compiled based on the answer to question no. 2, "Were you in a college prep program while in high school?" The analysis was made for each of three categories.

1. The total response collected (114 students).
2. The total response of students outside the area of English who were working on the computer (48 students).
3. The total response of students working on English programs (66 students). Many students did not respond, either because they were absent when the questionnaires were distributed; or they failed to return them.

Results of the Questionnaire

The grade expectation was higher outside English B (80 percent expected an A or B).

Only five percent of these students expected to fail whatever course they were taking. Of these five percent, all said they would like to use the computer again some time (no. 31). Some responses were a surprise. For example, 76 percent said questions are not presented too slowly. I had been very concerned that some exercises were too difficult, forgetting that the student could sit at the computer until he was ready to answer. The same percentage (62 percent) who said boring material could be interesting when presented by computer (no. 18) said interesting material was not boring on the computer (no. 22).

Some answers were disappointing. Students were obviously mixed in their preference for CAI over classroom instruction. Twenty-six percent reported they could learn more in the classroom.

English B students, as compared with others, do not seem to think the exercises take too long (70 percent said "No"); 76 percent do not think the computer makes the course more difficult. More English students said the programs relate to their class work (84 percent) as compared with those working on programs other than English (67 percent).

In the "No" column were several indirectly, as well as directly, positive responses from those who had not been college bound in high school. I decided to find the exact source of this support based on grades the students expected. My assumption was that the support may come from those who do not consider themselves above average students. If so, remedial programs could be the answer for those students on English who are striving for minimum levels of performance. The computer might just possibly be teaching these students.

Students who said they expected C or less in this course felt pushed by the computer (64 percent vs. 25 percent) and a greater need to talk about the programs when they returned to class (70 percent vs. 53 percent). In addition, they seemed more interested in the material (no. 18 - 89 percent vs. 69 percent). Finally, not one student in this group said that the course was a waste of time (no. 30) or that CAI in this course is poorly used (no. 24).

Several significant figures also appear in the columns for above average students (those who expected an A or B in the course), but they are not as impressive. For example, 67 percent said they would like to use the computer again. However, more of these students said CAI is de-personalized instruction (61 percent vs. 45 percent) and that the purpose of some programs was not clear (12 percent vs. 1 percent). The most glaring difference of opinion is apparent in questions 26 and 27. Only 12 percent of the better students thought that program errors detract significantly, as compared with 56 percent of the other group. However, the strongest criticism came from this group, too. Seventy-four percent said they prefer the classroom to CAI (compared with 10 percent).

Conclusion

None exists for the present. One major problem for these students in responding is that they have had no other programs to compare with those in English. In spite of all the obvious barriers to their evaluation, the students seem much more positive in their response to CAI than I anticipated. In any case, the target is the remedial student, and he appears to want help, regardless of the amount CAI may provide. In fact, it may be that the less secure a student is about his ability, the more he may need a continual tally and evaluation such as is possible with CAI.

The Value of CAI

Insofar as the value of the computer, I cannot comment whatsoever about cost effectiveness, perhaps the most important question for you. On our campus, the computer was there before CAI. To the question of whether CAI in English provides a valuable learning experience, the answer is both "yes" and "no." "No" because many students in remedial English are poorly motivated; many prefer sitting in a passive classroom situation; this is not possible with a computer. They cannot skip work by skipping class because the computer waits until they find time to work on it. Consequently, CAI is a burden and a responsibility; it also removes the comforting kind of contact with the instructor to the extent the student spends much of his time with an exacting type of measure of what he is doing. I would say "yes" to it as a valuable learning experience because it enables instructors to write programs tailored to unique student problems, even though the programs are limited by their ability to design them. I would say "yes" to it because the computer allows a kind of remediation wherein the student is not able to perform at a certain level. In most of the programs, for example, the first question is the most difficult rather than the most simple. If a student is not successful in the more difficult tasks, he is directed to a lower level of difficulty and later back to the more difficult questions. The answer is "yes" also because it rewards the more attentive student. Students are not in a classroom for fifty minutes, but at the terminal for however long they need to be. Furthermore, the computer shifts the responsibility of the course from the teacher to the student. He is no longer able to present the usual student complaints about not having heard the assignment...not having been there for the lecture. At the end of each program is his writing assignment, primarily to have it in writing, and so that he does not begin writing until he has gone through the theory involved in his particular assignment.

In an ideal system, the computer would understand English, of course. In addition, it would have:

1. A terminal which is quieter than a typewriter.
2. A maximum response time on all scanning routines of three seconds.
3. Extendable workspaces.
4. Specially designed type balls which would permit upper and lower case letters.
5. Sufficient storage capacity to accumulate and evaluate statistics related to student responses.
6. Some means of alleviating the monotony of program display. The student should be able to easily distinguish introductions, examples, specific instructions, and text. Possible answers to this problem might be varying the typography or presenting some information with audio cassettes.
7. Programming aids for language analysis—the ability to analyze a paragraph for syntax, grammar, and vocabulary. What such a system would require basically is much greater storage capacity and special routines, enabling one to define grammatical rules and lexical categories that could be interpreted to analyze sentence structure. The manipulating capability is already available in various compilers and compiler approaches developed for scientific application. The ideal learning situation would be one in which the student would not be assigned work in remedial English at all. He would be assigned to a regular freshman composition course. When he had problems, he would be assigned to specific computer programs designed to help him overcome them.

Program design in CAI presents a paradox. The programmer knows more about the limitations and potentials of the system, yet those programs most useful in English were designed by the instructor. Perhaps the programs are more germane when written by an instructor who knows little of the programming language than by a programmer who is unaware of objectives or instructional techniques. The programmer works for the system; the instructor designs his course for the student. "Science," in the form of a computer, may provide the answers, but "art," in the guise of the instructor, must still ask the right questions.

DRILLING SPANISH VERB FORMS ON REMOTE TERMINALS

Robert Phillips
Miami University
Oxford, Ohio 45056
Telephone: (513) 529-2733

Introduction

Two years ago, the Computer Center of Miami University added the time-sharing remote terminal language APL to its system, and a member of the staff suggested to me that I learn the language and develop something of pedagogical value for Spanish students. After some thought, I decided that there was a need for Spanish students to drill verb forms.

The verb is the heart of the sentence, and Spanish verbs have rather complicated morphology as compared to English verbs. A typical Spanish verb has about forty-two different forms, considering all tenses and moods. Because of this plethora of forms, some students have difficulty mastering them. And, if they cannot handle the verbs, they just cannot produce accurate sentences, either orally or written. Some of our students come from an audio-lingual background, in which most of their learning of verbs has been done orally; a few of these have never learned to write the verbs accurately. Others have difficulty learning the verb forms by repeating them in the language laboratory; they need some type of drill which they can do in a written manner. Still other students need more practice than that afforded by the exercises in their book or on the tapes in the language laboratory. And, some few students are frustrated by the speed of the tapes or the teacher, saying that they could get the verb form right if only the teacher or the tape would give them enough time.

Desiderata in Drill Programs

Before starting to write a program to provide the students with drill, I felt that there were five points which were necessary if it were to be successful:

1. Randomness. The drill items had to be presented to the student in random order, so that he would not learn the correct response in relation to the item preceding the one he was working on.
2. Immediate feedback. The student must be told immediately if his answer is right or wrong. Often, a student's problem is that he writes the wrong answer while doing his homework and does not find out for a day or two that it was wrong.
3. Ease of use for student. If the student had to learn anything about programming, or had to go through complicated computer ritual to use the drill, he might be turned off by it, considering it to be more trouble than it was worth.
4. One thing at a time. If the student were having trouble with regular past-tense forms, he would not be helped by being given irregular forms. Therefore, each drill should concentrate on one thing at a time.
5. Ease of use for teacher. From the beginning, I felt that the drill items would have to be changed as the students acquire new vocabulary and different complications in the verbs. To make this possible, it has to be easy for someone to make the changes.

The "Easy" Drill Package

With these five desiderata in mind, I wrote the program for a verb drill for the students. The time-sharing remote-terminal concept seemed right, because the student could work at his own pace, free from the tensions of time present in the classroom and the language laboratory. Furthermore, the remote terminal is very much like a typewriter keyboard, and virtually all students can use a typewriter, even if it is only on a "hunt and peck" scale. And, it is almost completely free of the paraphernalia of other computer applications: no job cards, no compiler parameters, no data description cards, etc. Once a program is developed and debugged, all the student need do is sign on, invoke the program, and then get to work on it.

I should add parenthetically here that I was greatly influenced in the design of my programs by a packaged part of the APL software. There is a mathematical drill called EASYDRILL giving the student practice in using APL's mathematical operators. As part of my learning of APL, I had used EASYDRILL and had liked several features of it. Therefore, some of the specifics of my drill program were copied from EASYDRILL.

I developed a package which I call EASY and which the student invokes by typing 'EASY.' (The student who needs remedial help usually wants something that is "easy." Since there is another package called HARD, which is discussed below, the student is assured that he is indeed getting the easy drill.) This program has nineteen different drills in it (each drill is on one "tense"), and the student chooses one at a time to work on. He can, whenever he wishes, change to any of the others. Each of the drills has twenty-five stimuli. There is also a "hint" associated with each of the stimuli. The student can ask for the hint at any time, or he can ask for the correct answer. If he does not know the answer, it may be better for him to get it, rather than waste time entering wrong answers, which he may accidentally remember.

When the student chooses the drill he wants to work with, he is given directions on what to do, and then a stimulus is presented to him. If the answer he types is correct, he is told that it is correct, and another stimulus is presented. If his answer is incorrect, he is told so and asked to try again. If his second response is correct, another stimulus is presented. If the second response is incorrect, he is told that he was wrong, the correct answer is typed for him, and the hint associated with that form is given.

One problem which occurred early in the developmental stage was the randomness of the stimuli. I had programmed it to give the stimuli in random order. However, I found that the student got annoyed when it repeated stimuli which they had already gotten correct several times. Therefore, I modified the program. Now it has in its memory a chart with twenty-five zeroes, one corresponding to each of the stimuli. When the student gets the answer correct on the first try, the zero for that stimulus is changed to a one. That stimulus will not be given to the student again until all of the zeroes have become ones. (When that happens, all of the ones change back to zeroes.) The student does not know that this is happening; he only knows that the "problem" forms come back again and again until he gets them right. There is one strange phenomenon that occurs, although I did not program it. The students believe that if they miss a certain form, that form is repeated after two or three intervening stimuli. Indeed, more than one student has told me that he "knows" that I wrote the program so that it would do that. It is apparently due to the fact that he rapidly disposes of the "easier" forms, and the program has to return to the forms which he has missed.

At any time during the drill session, the student can change to another drill or stop the drilling. Since there is nothing sequential about the drills, he does not interfere with any pedagogical sequence by changing when he wants. If he does want to change, he merely types 'CHANGE.' It types out the score he got on the drill he was doing, and then repeats the list of available drills. He then chooses the new drill he wants to work on.

It is quite easy for the teacher to change the stimuli and the answers for any drill. It requires typing in the twenty-five stimuli and the twenty-five answers, and the hint number associated with each stimulus. After each list is typed, the entire list is typed back so that the teacher can check it for correctness and make corrections where necessary. If the teacher wants, he can have it type out the entire drill: directions, stimuli, answers, and hints.

What the Student Does

This drill is very easy for the student to use. Let me describe briefly what he has to do. First, he signs on by typing an eight-digit number. The computer responds with a welcoming message. Then he types ')LOAD 1 EASYPAN,' which tells the system to load that particular package into its memory. Then he types 'DRILL,' which starts the drill program. It types a welcome to the Spanish verb drill and asks if he needs complete instructions.

If the student answers yes (by typing 'YES' and pressing the RETURN key), it types brief instructions, which summarize information that he has already read on a hand-out information sheet. It tells him to use accent marks and how to show them. (That was a problem: APL does not have accent marks. However, letters can be underscored, so we have an underscored letter stand for an accented letter.) If he makes a typing error, he is told to type an asterisk, press the RETURN key, and start typing the word over. It tells him to type 'PLEASE' if he wants a hint, or 'HELP' if he wants to see the correct answer. It reminds him to type 'STOP' or 'CHANGE' if he wants to stop drilling or to change to a different drill.

After the brief instructions, which take 1.3 minutes, a complete list of the nineteen drills is typed, and the student is told to enter the number of the one he has selected. After he presses the RETURN key, the directions for that drill are given. For one drill, the instructions say that a subject and an infinitive will be given; he is to type in the present indicative form corresponding to that subject. For another, it says that a verb will be given in the present tense, and he is to give it in the preterite form (one of the two simple past indicative tenses), keeping the same subject, etc.

After the stimulus is typed, the student types his answer, and gets a response from the computer. It is either "CORRECT" or "WRONG - PLEASE TRY AGAIN." If it was correct, another

stimulus is typed and he continues. If it was wrong, he tries again. If he misses the second time, it types "STILL WRONG - THE CORRECT ANSWER IS", the correct answer, and the hint to help explain it.

It goes on for as long as the student wants to continue with that drill. The important thing is that the student can continue working at his own pace for as long as he wants. If he wants to spend three hours on one drill, he may do so, providing terminal time is available. When the student types 'STOP' or 'CHANGE,' it types out his score, giving him the number of stimuli he attempted, the number right on the first try, the number right on the second try, and the number he failed. It disregards those for which he asked for help.

From the student's point of view, then, this is very simple to use. He has to be able to type the forms correctly, but if he does make a typographical error, he can correct it simply. The student does not have to understand anything at all about the way in which the program is worked. He is the sole judge of when he has drilled enough, and he works at his own speed.

The student does not realize it, but at the time that he is drilling on one particular tense, he may be getting passive drill in another tense. In all of the drills but three, the stimulus is another verb form, generally the present tense. We Spanish teachers find that the students often look at a verb form and do not pay any attention to what the subject of the verb is. This is due to a difference between English and Spanish: English must have an explicit subject, while in Spanish the ending of the verb tells what the subject is. Thus, when drilling the future tense, the student must correctly identify the subject of the present-tense stimulus. He may, for example, look at the present-tense verb, assume that the subject is "I" and type the future form accordingly. When he gets it wrong because the subject was supposed to be "he," the student is forced to take another look at the stimulus and to identify the subject correctly.

The "Hard" Drill Package

I had some students in more advanced Spanish classes who tried the drill package described above, but who found that it was too easy and too restricted in what it offered. They wanted something which would present them with various tenses and a greater variety of problems than EASY has. I decided that to be worthwhile, a more complex drill would have to give the students drill with at least forty different verbs, in most of the tenses, and with all subjects. Such a drill would involve a total of 2,800 different forms, and I was not about ready to type in 2,800 stimuli and 2,800 answers. Furthermore, there is not enough memory storage available in APL for that many words. I wondered if it would be possible, instead, to give the computer stems and endings, and then program it to generate the forms.

This intriguing problem was one which the innate programmer in me immediately had to solve. I did write a drill which does specifically that. It has in it fifty different verbs of every type: the completely regular ones, those which are slightly irregular, those which are very irregular, those whose stem changes form, and those which have to change spelling when certain endings are added. This program stores the stems of the verbs and all possible endings. It has some very complicated tables which tell it what stem to use for each particular subject in each particular tense, and what ending to use for that subject in that tense. Thus, each time the program calls for a verb form, the program itself generates the form: no complete forms are stored anywhere. In my final design, there are fifty verbs, nine tenses, and eight subjects for a possible total of 3,600 forms.

For the stimulus, the terminal types out an infinitive, a subject, and the name of a tense. The student is then expected to type in the correct verb form. The actual answer is not given by the computer unless the student misses it in his two tries. The student does not see the computer generating each of the forms from the stems, endings, and tables, but this process is done twice each time through: once to generate the infinitive (no complete forms are stored) and once to generate the correct answer.

Before the program could be written, it was necessary to arrive at some type of analysis of Spanish verb morphology. Although I quite firmly believe that Spanish verbs consist of three "slots" (the first being the "stem," the second being the "them vowel," and the third being the "person-number (subject) indicator"), I decided, for practical programming purposes, to divide the verb into two parts: stem and ending. One reason for this is that a "three-slot" analysis has to make use of a number of "zero" allomorphs. That is, the slot is empty in that particular form. If one were to use the empty slots, the program would have to look for empty slots and then ignore them. Although quite possible to do, it would make the program more clumsy. Since the purpose was to generate verb forms rapidly - and not to give lessons in the theory of Spanish verb morphology - I decided that it would be better to have an "elegant" program based on a more "rough and ready" analysis of verb morphology.

Another problem was what to do with the perfect tenses, since they also had to be drilled. The perfect tenses consist of a form of the auxiliary verb haber plus the past participle. The

past participle can be formed as a stem and an ending (although with difficulty in the case of irregular verbs), but then one would also have to take a stem and an ending for the auxiliary haber, which has to show tense and subject. I solved this problem by calling the entire past participle the stem and the auxiliary verb (in its complete forms) the ending, and then putting the "ending" in front of the stem, with a space separating the two. Certainly an inelegant solution from the point of view of Spanish verb morphology, but one which fitted the needs of the situation quite well.

For students working at the more advanced level, I felt that working with only one tense at a time was unnecessarily restrictive. However, it would not be right to force the student to work with all nine tenses if he wanted drill in only one or two. Therefore, the program was written to allow the student to work with whatever tenses he wants. If he wants to drill only one tense, he can do so. Or, he can work with two, three, four, or up to all nine. Part of the "preliminaries" consists of having the student select the tense(s) he wants to drill. From then on, he works only with those tenses. Thus, he can drill one troublesome tense, review a group of tenses, or review all of the tenses at one time. The one thing that is not possible for him to do is to select a certain type of verb (such as irregular futures, irregular preterites, etc.) to work with. He has to take all types of verbs with the tenses he has chosen. This, however, is apparently not felt to be a lack by the students, for none has ever asked me if it would be possible to sub-divide the verbs by class.

This program, which is called HARD, is modeled on the EASY program described above. It was not possible to have a list of hints. The programming of hints would be so exceedingly complex (it would involve a three-dimensional table of numbers which would be fifty by nine by eight) that I decided to forego the hints.

The package has the advantage of never needing any changes by the teacher. Since there is a total of 3,600 different combinations of verbs, subjects, and tenses, the student can drill for a very long time without feeling that he has exhausted the possibilities of the drill.

From the student's point of view, this program works very much like the EASY drill outlined above. It gives him simple general directions, then types out the nine available tenses (present indicative, imperfect indicative, preterite, future, conditional, present subjunctive, past subjunctive (-ra form only), present perfect indicative, and past perfect indicative). He types in the number(s) corresponding to the tense(s) he wishes to drill. From then on, the stimuli consist of an infinitive, a subject, and a tense. The student has two tries to get it right; if he does not get it right on the second try, the correct answer is given to him, so that he can learn it and analyze why he was in error. Although he cannot ask for a hint, he can ask for the correct answer by typing 'HELP.'

When the student wishes to stop, he types 'STOP.' As in the EASY drill, it types out the number he attempted, the number right on the first try, the number right on the second try, and the number failed. In addition, it breaks these data down by tense, showing him how many he attempted of each tense, how many of those were right on the first try, etc. In this way, the student can see where he is having problems, and concentrate on those areas the next time he drills.

Operating Considerations

Although the terminals are easy to use, and APL is not complicated, there were a few difficulties which annoyed the students, and which had to be overcome if the drills were to have maximum pedagogical value. The first was the matter of typographical errors. To correct an error in APL, you must backspace to where you made the error--and it's hard to see exactly where that was with the IBM type-ball--and press the ATTENTION key. Then you resume typing from that point. Students get confused by how far back to backspace, and don't know what to do if they go back too far. So, I programmed a rather easy way to do it: the student types an asterisk, presses the RETURN key, and starts the answer all over. They find this considerably less complicated.

Another problem which occurred was with the terminal itself. Our terminals have plastic-type-shields which cover the first couple of lines on the paper. While one can see through these shields, they are not easy to read through and they annoyed the students. Therefore, when it types the stimulus, it advances the paper two extra lines, so that the stimulus is above the type-shield. The student can see it easily.

Perhaps the gravest difficulty was with the randomizing of the stimuli. While the APL system has a very fast, very easy random-number generator, it has one grave fault: it uses the same seed each time the workspace is loaded. You can picture the generator as being an infinitely-long string of numbers which is always the same. That means that you get the same random numbers in the same sequence each time you use it: that is not random! If the students were to use different tenses each time they used the drill, there would be no repetition noticeable to them. However, I had a couple of students who always used HARD drill to drill all

nine tenses. They discovered that each time they used the program, the same subjects, verbs, and tenses came up together in the same order. Clearly, if they were to get the full benefit of the drill, I had to make it so that the random numbers were indeed random, rather than predictable. This I did by having the program get the time of day in milliseconds from the central processor, convert that into hours, minutes, seconds, and milliseconds, and then use those numbers in a loop which asks the generator for random numbers. This loop gets a random number of numbers from the generator, but does not print them out; sometimes it will ask for only five or ten, it may ask for over a thousand. Thus, running the program through this loop in effect starts the drill at a different place in the string of numbers each time it is used. Again, the student is not aware that this is taking place--the processor is doing it while the terminal is typing out the instructions. Both EASY and HARD have this randomizing routine.

Results

What has been the student reaction to these drilling programs? First, we have quite limited terminal time. (There are only eight terminals in the university, and they are available only three hours per day, giving only twenty-four terminal hours a day, five days a week.) Students are limited to thirty-minute periods on the terminals, meaning that only forty-eight students can use them each day. Our Spanish students have to compete for time with students in Systems Analysis, Math, Physics, Business, Psychology, Sociology, Chemistry, Political Science, and Education Psychology. Furthermore, the system is available only from 4:00 pm to 7:00, which covers the dinner hour. Because of these limiting factors, we made announcements in classes that the drills are available, but no student was obligated to attend. Nonetheless, the students who elected to use this system have been quite enthusiastic about it. During the first quarter of 71-72 academic year, we logged approximately forty hours of terminal time by Spanish students, which means approximately eighty different uses by students. Very few first-year students were using it, because they had had only the present tense during most of the quarter. Our prediction is that it will be more widely used by them during the second quarter, when they have to learn five more tenses.

During the winter quarter of the 70-71 academic year, we were able to reserve twelve "slots" per week, and usually had eleven students sign up for those times. These were all first-year students, studying those new tenses. The HARD drill was not then available for the advanced students.

All in all, I feel that our students are making good use of the terminals. If we had greater availability of terminals and terminal time, we would have more use. However, if too many more students want to use the terminals now, we simply will not be able to accommodate them with the resources at hand.

In use, the students seem to enjoy doing the drills, not finding them to be so boring as they profess to find the oral drills in the language laboratory. Many of the students say that it is "fun" to work on the terminals, matching wits with the computer. None seem to be bothered by having to type their responses. My experience has been that the students are awed by the mechanical aspects of it for about the first five minutes (the terminals type at fifteen characters per second, which is admittedly awesome), and feel slightly uncomfortable during that time. After the first five minutes, however, they adjust very well to the situation. We find a very high percentage of students who come back for more drill.

How much can a student do in the thirty minutes allotted to him? It varies greatly from student to student. Those who are very slow will do about twenty-five stimuli. This means that in the EASY drill, they won't even cover all of the stimuli one time through, unless they get them all right on the first try. More common is about forty stimuli. Last year, one girl regularly did between ninety and one hundred stimuli on the HARD drill. But she almost never made a mistake and could type fast, besides. Most students do not feel that thirty minutes at one sitting is quite enough. I suspect--but have no way of knowing--that over an hour would lead to a situation of diminishing returns.

Do the students learn from this? My answer is definitely yes, but I do not have concrete data to prove it. Because of the limitations, it has not been possible to do formal experiments, using control groups, etc. In the very early stage of development I used one of my first-year students as an experimental subject. I had given his class a quiz on a certain tense, and he had missed nine out of twenty-five items. He then drilled on that tense for over an hour on a Friday evening. On Monday, I gave another quiz on the same tense, and he missed only two out of twenty-five. Fewer than half of the forms on the quiz were included in the drill: he had learned the process of formation, and not just memorized certain forms.

In my third-year composition class there is no review of verb forms. I simply tell the students that I assume they know all of the verb forms. After the first composition of the quarter--on which there were numerous verb-form errors I announced the availability of the HARD drill and told them where to get the information sheets. Again, no statistics (we do not keep

records in our Department, since we do not have a terminal), but the number of verb-form errors on compositions has greatly diminished. Doubtlessly, many of the errors on the first composition were due to the students' having been away from Spanish during the summer. Nonetheless, enough students have told me how helpful they have found the drills to make me certain that computer-assisted drill in Spanish verb forms has a valuable place in our university.

The Future

What is the future for this type of drill? I believe that it is definitely here to stay as a valuable adjunct to classroom instruction. I would like to see the drills expanded to cover other foreign languages as well. I believe, too, that the concept of the drills can be extended to items other than verb forms. While I do not envision the possibility of having the computer check a student's translation from English into Spanish, it could be used with certain selected topics, such as the use of ser versus estar, the use of the preterite versus the imperfect, and, perhaps, even the use of the subjunctive in complete sentences. Naturally, each type of drill will require a considerable amount of thought on the pedagogical goals, and then on how to achieve those goals in a program.

Our greatest need now is for more facilities, so that the remote-terminal time-sharing system can be on-line at all times. As it now stands, our language laboratory is a valuable help to the students. I would like to see one terminal on-line at all times in the language laboratory. Thus, the students could come to drill in the language laboratory in various ways: orally for those things most appropriate for oral drill, and on the terminals for those items which can be profitably drilled and reinforced by writing and repetition.

ACKNOWLEDGEMENTS

The author expresses gratitude to E. Arthur Fiser, Director, and to David Probert, Consultant, both of Academic Computer Services of Miami University, for their constant encouragement and help, and for their unflagging dedication to the principle that the computer should help all segments of the university community, including foreign languages.

COMPUTER-AIDED GRAPHICS AS AN ART FORM FOR THE ARTIST

Grace C. Hertlein
Chico State College
Chico, California 95926

Although course development for the teaching of Computer Art for Artists began at Chico State College in April of 1970, actual teaching of the course did not begin until September, 1971. The course was offered in the fall of 1970, tailored to the needs and interests of artists, yet it failed to reach minimum enrollment. Other institutions have invited artists to experiment with computer-aided creation and have experienced the same hesitancy on the part of the artist to use advanced technology to create works of art. A review of the literature regarding art and technology, and perusals of avant-garde technological exhibits reveals heavy use of industrial materials and procedures of fabrication, yet a low level of complex machine-aided creation. Thus computer-aided creation represents the apogee, at this moment, of complex machine-aided creation. When viewed from this attitude, it is not surprising that the artists are reluctant to leave the known manual world in which they have confident capacities.

At first this appeared ironic to the instructor, an artist "converted" from the ranks of fine art, one who understood clearly the difficulties that the humanist experiences when confronted with the complex and demanding computer.

Our course description for artists is as follows:

1. Investigation and use of computer graphics as an art form for the non-programmer (artist);
2. Use of computer graphics languages combined with art techniques familiar to the artist, applicable to the creation of computer art graphics;
3. Development of design planning, execution of computer graphics leading to a personal style in computer art;
4. Applications to painting, sculpture, weaving, printmaking and applied design.

Prominent writers, including Holland, Milic, and Mesthene, have detailed the problems that the humanist faces in his first confrontation with the computer. The instructor in this course, an artist-teacher, with long experience in fine art and in the teaching of art, had experienced first-hand the psychological difficulties that the artist encounters within initial computer-aided experimentation.

The plan was to take the known world of art, creativity, material usage, design proficiency, and bring it to the unknown world of the computer. In this manner, one proceeded from the known to the unknown.

Course objectives are:

1. To develop an acceptance of computer graphics as an art form by an exposure to computer-aided creation;
2. To afford in-depth experimentation of machine capabilities, relating to known art techniques and unknown computer techniques, affording early proficiency in the creation of computer art;
3. In-depth study and use of art materials, design techniques, using simple programming as the basis for experimentation.

It appeared during our first year that the artists were unwilling to "jump the crevasse" of the unknown, even with a sympathetic and qualified guide.

Interest on the part of Computer Science students in computer art was very high, and a generous listing of science students awaited entry on an additional reserve list. In the fall of 1971, a group of seven artists signed up for the course. This presented a problem, in that a large number of Computer Science students applied for the same time slot. In order to accommodate both groups, on an experimental basis, we put artists and programmers together in our evening course, with the provision that each group would help the other. This also meant extra tutoring for the artists by more experienced programmers, and a faster introduction into more sophisticated programming than the artists would have encountered in an isolated artist group.

Originally we had planned to use an industrial drafting program for the artists. This language is easier to comprehend, and there are built-in software capacities, such as rotation, mirroring, etc. that afford moderately sophisticated art graphics within a few weeks. The instructor had used this system in early research, and had also made use of this easier language in the first pilot group of programmers. At that time, experimentation with the first programming group revealed a larger variety of more artistic and varied graphics when using this system. It also revealed, when contrasted with Fortran, that the highly advanced programmers scorned such an "easy" language and disdained its usage as beneath their technical capacities!

However, with installation of a new computer system, the drafting system compiler proved excessively inefficient time-wise, and the decision was made to use Fortran with the new group of artists, using tutorial help from programmers.

There are many institutions desirous of teaching such a course, and many await the "ideal" system, the "ideal" language. In the midst of economic realities, the colleges and universities find that if they wish to pioneer, they must make do with what they have, and just begin.

This is what we did. We were cognizant of the advantages and disadvantages of such a procedure, but felt that more was to be gained than lost.

The question then might be, what is the ideal artist system?

1. A digitized sketchpad or digitizer;
2. CRT lightpen and/or joystick input;
3. Easy programming languages designed for non-programmers.

Designers, artists, humanists, all cry for an easier mode of input. They do not wish to learn a second career. Yet cognizant of this need, the above "ideal" system has disadvantages:

1. The artist often transfers his manual art to the machine and fails to explore the innate nature of the material or media, in this case, the computer. He then fails to discover the uniqueness of the media. This "truth of the material" is more readily discerned via sophisticated programming and mathematics.
2. The artist repeats things on the computer that might better be accomplished manually, and the resultant output resembles a bastardization of manual art, via the computer!
3. The artist fails to appreciate the uniqueness of the computer, and to perceive the wedding of art and technology, which is a union of science and art.

Thus, easier modes of input result in easier output of initial graphics, but at a lower level of technical proficiency. Interdisciplinary means the union of two disciplines, i.e., the practitioner learns two disciplines, and combines them into a balanced whole. And in computer art, after the initial exposure, we are always aware of the level at which the computer is being used.

Our small group of artists were isolated once a week for two hours with an assistant in a laboratory session. They started from scratch:

1. What is a computer? How does it work? What can it do?
2. What is a program? How does one write a program in FORTRAN?
3. They learned to keypunch, and to debug their own programs.
4. They used very simple practice exercises to gain proficiency in programming, to gain accuracy, and to develop confidence in their new capacities.

During the other three hours a week, they met with the programmers for lectures, films, laboratory programming and sketching sessions. Our mixed group of artists and programmers fared fairly well in many ways. Informal in-depth discussions between artists and programmers regarding difficulties helped both groups to adjust to new interdisciplinary creation. Problems encountered were:

1. Programming Difficulties. The artists were not mathematically oriented, and even with help from programmers, they experienced difficulty in using FORTRAN. We used point to point programming at first, then advanced to more symbolic input. We played substitution games, so that symbology was merely using something for a known something else. Programming then became game-like and more enjoyable.

2. Thinking Machine in Designing Components. Both groups had difficulty in designing for the computer. Laboratory sessions, blackboard illustrations by the instructor, and sketching sessions clarified this concept. The reiterative capacities of the computer were exploited, and regular progressions of increment or decrement were an integral part of designing. The "esthetic" of the computer became a goal in designing and programming.
3. Interdisciplinary Weaknesses. Both overcame their initial feeling of inadequacy. Each group was encouraged to base their new work on their dominant strength, and to progressively grow in the new discipline.
4. Verbalization of Ideas. Programmers and artists alike experienced difficulties in written and oral verbalization. While programmers had difficulties in initially analyzing subjective and objective ideas, their scientific training afforded faster growth in such analysis. Artists, on the other hand, through "mental programming" had relied upon intuitive, emotive, non-analytical methods to create, and their verbalization and analysis was more difficult than that of programmers.

In addition to these difficulties, our beginning classes were all running preplanned mode, although advanced classes had the option of "hands-on" running. It is the writer's strong opinion, that if at all humanly possible, computer art should be run in the "hands-on" mode:

1. Students require the emotional involvement of seeing ink applications develop on varied art papers;
2. They need to see for themselves where the design may be overdeveloped;
3. Myriad ideas for development are perceived when the artist runs his own graphics, as natural variations from first creation. The whole purpose is to "see" as you are working, and not merely to intellectualize theories.

To compensate for this preplanned mode, we introduced randomization earlier to this mixed group, including the artists. We also used known, tested "design states" for varying graphics, using examples of student-teacher work, slides, blackboard illustrations and group brainstorming sessions. With these methods, students were able to vary their components and achieve more sophisticated graphics than they would have, had they "discovered" such states in "hands-on" running of their work.

Great emphasis was made of the concept of recognition of known parameters, whether they were equipment, languages, or one's own limitations. The ideal world is never ours, yet increasingly, by cognition of where the limit occurs, one may affirmatively create within these limits. To this beginning group of mixed students, preplanned mode was a limitation, yet we introduced more advanced concepts to compensate for such a seeming hindrance. Within the next semester, these same students will focus on "hands-on" running modes, participation programming and running states, painting techniques, all of which are not possible in preplanned modes. Yet by contrast, these same students examined graphics that were preplanned, randomized, painted, block-printed. Randomized graphics, with stated limits, often emerged as their preference, revealing a more "natural" variation of the subject than regular preplanned, participant programming or painting techniques. In other words, every mode of programming and running, every system has advantages and disadvantages.

In summary, cognizant of the advantages and disadvantages under which this group operated, artists learned more technically about the computer:

1. The artists learned the mechanics of programming, keypunching, debugging, editing programs.
2. They experienced new insights into contemporary art, and more, that of complex machine-aided creation, by doing, and not just reading about it.
3. They learned to program fairly well in FORTRAN, and to symbolically perceive mathematical relationships that were moderately advanced for such a group.
4. They perceived a dimension of logical thinking, which they began to apply to other areas. Some felt this alone was worth the travail they had experienced.
5. Programmers developed multi-purpose (more complex) subroutines for artist use, and the artists learned to understand more complex programming in this manner.
6. Frustration was overcome gradually. Students felt a great sense of achievement in their esthetic and intellectual accomplishments.

7. Affirmation of technology per se, as a part and parcel of our society resulted from this exposure.
8. Student artists learned to appreciate more fully the ability to analyze and define an artistic problem, and to seek varied and alternate solutions to such a problem. In other words, they gained fuller control of their own creative processes.
9. The majority of the artists felt they had passed the period of difficulty, and plan on taking the advanced course, to continue their computer creation.
10. The artists emerged with a new intellectual independence, i.e., they had learned a new way of thinking, which liberated them from relying totally upon intuitive processes alone to create. They now had not only control, but more sophisticated means of achieving creativity.

It is obvious from the foregoing that the artist experiences greater psychic difficulties in "going interdisciplinary" than does the programmer. The world of science is more demanding than the world of art. This brings up the interesting question: "Is it easier to learn to create art via manual modes than it is to learn to use the computer in a moderately sophisticated manner?"

The answer is that it is far easier to learn to paint in watercolors, to sculpt a head in clay, than it is to learn the complexity of the computer.

But a far more interesting concept is this: It is moderately easy to take a non-programmer, a non-artist, and to teach this student how to program, and simultaneously how to create, using the computer as an aid in creation. We rely on the innate creative capacities of the human being, merely releasing them using experimental techniques.

Thus using the computer cannot only be a complex learning situation, it may be coupled with the teaching of art and the creative processes.

USE OF THE COMPUTER IN INTRODUCTORY ALGEBRA

Thomas Ralley
The Ohio State University
Columbus, Ohio 43210
Telephone: (614) 422-5710

This is a description of the use made of the computer in an introductory algebra sequence given at Ohio State. The algebra involved is well known and several of the procedures described have been done by hand for many years. While a good deal of energy has been devoted to the question of how computers might be employed for research in algebra (see [1], [2]), there seems to be a dearth of information about ways machines can be employed in an undergraduate algebra course. There appears to be great potential in this area and it is likely that interesting work has been done here which has gone unannounced or unnoticed. We hope that this paper might stimulate more discussion about past results and future plans.

The sequence described is three quarters work covering number theory, linear algebra and abstract algebra. By treating this as a full year sequence rather than three separate courses we hope to avail ourselves of opportunities to interweave the abstract with the concrete in a natural manner. Utilization of the computer is another step in this direction.

The students involved are primarily juniors majoring in mathematics. Their career goals after a bachelor's degree are widely varied; medicine, law, sociology, industry, and indecision. Perhaps one fifth are interested and capable of doing graduate work in mathematics (most undergraduate math majors who might be expected to go on to graduate school are identified earlier and counselled into the honors sequence) and so no pretense is made of giving a graduate preparatory course.

The system used was CPS (conversational programming system) with an IBM 360/150. Programs were written in the CPS PL/1 language. Communication with the machine was via IBM 2741 terminals--in appearance, a typewriter connected to the computer by telephone. For the beginner, CPS offers some significant advantages over punched cards. As the name implies, while the student is typing a program, he is engaged in a conversation with the computer. If an instruction is incorrect, the computer replies immediately with an error message. If the programmer wants to alter the program while it is running, he may stop execution and do so. These and other options make it especially attractive for inexperienced individuals who are prone to ensnare themselves in numerous programming errors.

In a venture such as this, where no previous experience with computers is required, one serious difficulty is teaching students to program. Fortunately about half the class had some previous training and several were quite expert programmers. Experience was advantageous, for essentially we adopted the attitude that programming is a skill one learns by doing. After discussion of the basic commands and a few illustrative examples, students were given a fifteen page handout containing a description of the most commonly used PL/1 commands and conventions and were then left on their own. From time to time common difficulties arose which were discussed in class.

Student response at the end of the quarter indicated that this procedure left something to be desired. Apparently more care needs to be given to the content of the instruction sheets and more class time devoted to discussion and illustration of fundamental techniques. But no matter when an individual begins to program, he encounters frustration. Since we intended to make use of the machine throughout the year, one quarter for adjustment seemed reasonable.

Students were asked to write four programs during the quarter with a fifth program optional. Although programs were not counted in determination of the final grade, the class was asked to turn them in for checking. About 80% of all assigned programs were completed.

As an introduction to the machine, the first program was to determine all primes between one and two hundred. This is a rather nice problem in that there are a number of ways in which it can be done, ranging from extreme brute force to relatively sophisticated sieves. Most programs relied more on brute force.

Second was the old classic, computation of the gcd of two given integers via the Euclidean algorithm. Carrying this a step farther, the third program asked for a solution to the diophantine equation expressing the gcd as a linear combination of the given integers a and b . Interestingly enough, rather than the usual procedure of unraveling the steps of the Euclidean algorithm, it turned out to be preferable, for programming ease, to determine the convergents in the continued fraction expansion of a/b (see [3]).

The last program was to determine the elements in $\mathbb{Z}/(m)$ having multiplicative inverses. This provided a 'practical application' for one of our classroom theorems and also a family of examples to be drawn upon later in our study of groups.

Related to this was an attempt to use the computer as a Gaussian pencil, furnishing new data from which new discoveries can be drawn. A program giving the value of Euler's phi function for integers between one and five hundred was stored in the machine. After typing in various values of m and obtaining the values $\phi(m)$, students were asked to conjecture a general formula for $\phi(m)$. This was not as effective as had been hoped. Most students failed to observe the multiplicative nature of ϕ despite a number of strong hints. Apparently a bigger push in the right direction was needed.

A fifth program, finding the Cayley table for the group of units in $\mathbb{Z}/(m)$, was optional. Because of the difficulties in formatting the output, this presented the greatest problem, even to the best students.

On a few occasions students seized the initiative in using the machine. One group became interested in pythagorean triples and presented a report of the class which included a program they wrote to find all primitive triples with certain bounds.

Our plans for the current quarter also include the computer. In class, we expect to do a bit more group theory, study polynomials and then begin linear algebra. Programs we anticipate assigning during the quarter are as follows:

First, determine the orders of all elements in the group of units in $\mathbb{Z}/(m)$. By utilizing earlier work, this program should be easy to write. It should provide experimental data that students can use in considering the relationships between orders of elements and the order of the group.

Second, determine the gcd of two polynomials in $\mathbb{Q}[x]$, where \mathbb{Q} denotes the field of rationals. This should help draw the analogy between integers and polynomials with coefficients in a field. In addition, programming the computation might help some students recognize the general form of coefficients in products, something that might not have occurred in their earlier exposure to polynomials stressed only numerical computations.

Third, determine all monic irreducible polynomials of degree less than m in $\mathbb{Z}/(p)[x]$ for given prime p and given m . Here we expect to make use of the exclusion principle - using the lists of monic irreducible polynomials of smaller degree to construct all reducible products, then excluding these from the list of all monic polynomials of the fixed degree. These irreducibles will be of value later when we study field extensions.

Finally, determine the Hermite normal form of the augmented matrix for a system of linear equations with coefficients in the reals (sometimes called the Gauss reduction process). Systems of linear equations occur in so many contexts in linear algebra that the utility of the program is quite apparent. In addition, past experience has shown that some students never recognize the fact that there is a systematic procedure for finding solutions of these systems of equations. Programming the procedure should impress its algorithmic nature upon their minds.

In summary, our use of the computer has had several functions. First, the student gains a better understanding of material when he has to write a program. In this sense, programs have the same purpose as problem sets. Second, the computer acts as a Gaussian pencil, providing experimental data from which students can increase their fund of examples and make conjectures. Third, the computer serves as a labor saving device, performing computations that would be tedious and time consuming if done by hand.

Initially some students (primarily those without programming experience) found working on the computer a bit distasteful. However, we feel that the pedagogical value, together with the fact that experience with computers is an asset in the competition for jobs under the present economic conditions, makes the frustration and labor on the part of the student worthwhile. Despite their reservations, students have been willing to try programming and by and large we feel that our use of the computer has been successful.

REFERENCES

1. Computers in Algebra and Number Theory, SIAM - AMS Proceedings, volume 4 (1972).
2. Computational Problems in Abstract Algebra, edited by J. Leech, Pergamon Press, 1970.
3. An Introduction to the Theory of Numbers, I. Vinogradov, Pergamon Press, 1955.

1. Program for finding integers x and y such that for given integers a and b (with $0 < b < a$) $ax + by = \text{gcd}(a, b)$

```

get list(a,b)
declare q(*) /* here * is taken suitably large */
l=1
nex: a(1)=a
b(1)=b
q(1)=trunc(a(1)/b(1))
r(1)=mod(a(1),b(1))
if r(1)=0 then go to con
l=l+1
a(1)=b(1)
b(1)=r(1)
go to nex
con: R(0)=1
S(0)=0
R(1)=q(1)
S(1)=1
do k=2 to l-1
R(k)=q(k)*R(k-1)+R(k-2)
S(k)=q(k)*S(k-1)+S(k-2)
end
if a*S(l-1) - b*R(l-1) > 0 then put list(S(l-1), '- ', R(l-1))
put list('- ', S(l-1), R(l-1))

```

2. Program for finding the units in $\mathbb{Z}/(m)$

```

get list(m)
declare U(*) /* here * is chosen suitably large */
l=1
U(1)=1
loop: do k=2 to m-1
x=m
y=k
gcd: r=mod(x,y)
if r=0 then go to check
x=y
y=r
go to gcd
check: if y>1 then go to next
l=l+1
U(l)=k
next: end
do i=1 to l
put list(U(i))
end

```

A TIME-SHARING COMPUTER IN THE DIFFERENTIAL EQUATIONS COURSE

Allen D. Ziebur
State University of New York
Binghamton, New York 13901
Telephone: (607) 798-2147

Introduction

Since mathematics deals with numbers, it is obvious that the digital computer has a role to play in teaching it. To many people, that role is simple; let the machine grind out answers to "practical" problems. This article describes a course we have developed over the last half-dozen years, in which we stress a different way to use the computer. Our machine does grind out answers to problems (we are not interested in the programmed learning aspects of CAI), but these problems are not supposed to be "practical." As in any math course, our problems are designed to deepen the student's understanding of the theory we are trying to expound.

We therefore think of the computer as a device for teaching fundamental concepts. For example, a function is a set of pairs of numbers, and a machine can spew out hundreds of pairs of numbers in the twinkling of an eye. An integral is a number, and because a computer can integrate "anything" a student should get the feeling that "anything" is integrable. (This remark isn't as foolish as it sounds. How many students come out of a standard calculus course believing that the equation $f(x) = e^{-x^2}$ defines a non-integrable function?)

In many ways, a course in differential equations provides an ideal setting to test this philosophy of the computer's role as an educational tool. Traditionally, it is here that a student "really" learns his calculus, and the subject provides many opportunities to call on the computer to aid in this endeavor. We have interpreted the content of our course quite liberally. Before solving differential equations, we solve some numerical equations, thus giving us a chance to bring in Newton's Method, for example. And since many differential equations are solved by integration, it is natural to introduce Simpson's Rule, and so on.

As for differential equations themselves, their solutions are functions, sets of pairs of numbers. It makes a student really believe existence theorems when he uses, say, the Runge-Kutta Method to generate a set of pairs of numbers that constitutes a subset of the solution of a given initial value problem and realizes that he could do the same thing for "any" initial value problem. We define certain functions, such as Bessel functions, as solutions of initial value problems. The trigonometric functions could, of course, be defined in the same way, and a student who produces a table of values of the Bessel function J_1 by the same techniques he has just used for the sine function is less likely to stand in awe of Bessel functions than is one who thinks that mathematical tables are delivered by the stork.

Practical reasons also argue for introducing the student to the computer in the differential equations course. For one thing, it has a smaller population than the calculus course, so less machinery - administration plus hardware - is needed. Furthermore, there isn't a lot of extra time in the introductory calculus course, and it's hard to fit the computer in.

Some Details

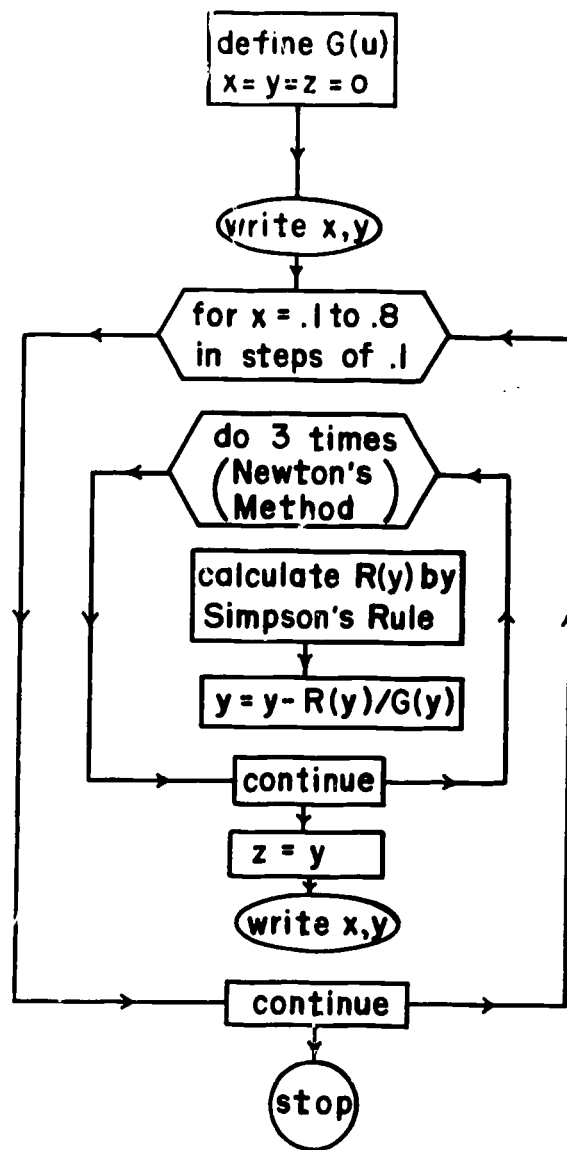
What made our course possible was the invention of the time-sharing computer, and we have used several time-sharing systems over the years. In 1965-66, we had a Teletype terminal tied in by long distance phone lines to the Dartmouth computer. With its beautiful language BASIC, that system was simplicity itself, and students were writing meaningful programs with only an hour or so of instruction. Then we used a remote access system (RAX) with FORTRAN on our own Model 360/40. To my surprise, its complicated software and hardware (IBM 1050 terminals) didn't seem to bother the students. Later, we had a much simpler time-sharing system (ITP) on a Model 360/67, and we shifted back to BASIC. We have also run the course using APL. Actually, our programs are so simple that the particular computer system being used doesn't seem to make a great deal of difference. Familiarity with the computer is not a prerequisite for the course; we start from scratch.

Perhaps because it is my native tongue, I prefer BASIC. It and FORTRAN are so close to ordinary mathematical notation that little time is wasted learning them. The same cannot be said for APL; one has to count on sacrificing a good deal of time to teach the language if he wants his students to write programs of their own. (And the conflict between the notation and terminology of APL and mathematics doesn't add anything to the learning process.)

Specifically, our one-semester course contains the following computer-related topics, handled more or less as you would expect them to be:

1. Solving numerical equations (using, for example, Newton's Method)

327



```

G(U)=1/(1+U*U)
X=0
Y=0
Z=0
WRITE(6,1) X,Y
1 FORMAT(IX,F3.1,F10.4)
DO 2 I=.1,.8
DO 3 J=.1,.3
S=G(Z)+4*G(.25*Y+.75*Z)
+2*G(.5*Y+.5*Z)+
4*G(.75*Y+.25*Z)+G(Y)
R=-.1+((Y-Z)/12)*S
Y=Y-R/G(Y)
3 CONTINUE
Z=Y
X=X+.1
WRITE(6,1) X,Y
2 CONTINUE
STOP
END
  
```

FIGURE 1

2. Functions and functions defined by integrals (such as \sin)
3. Initial value problems solvable by integration (linear, variables separable)
4. Existence theorems (based on Euler-Cauchy polygonal approximations)
5. Making tables of values of functions defined by initial value problems (using Runge-Kutta, Milne)
6. Series solutions of differential equations (trigonometric functions, Bessel functions, Legendre polynomials).

These topics are interspersed with other standard material of differential equations, for which the computer is used only slightly, if at all. Thus, we also treat systems of differential equations, linear differential equations of the second order, and so on. In general, computer-related exercises are used when they seem natural, pencil and paper exercises when they seem more appropriate. More details are given in the book [1] that grew out of the course, which happens to cover essentially those topics listed in [2, pp. 33-34].

An Example

This example is one of our most complicated ones, but it serves to illustrate what we are trying to do.

Example. Solve the initial value problem (1) $y' = 1 + y^2$ and $y = 0$ when $x = 0$.

The given differential equation is separable, so our theory tells us that the solution of problem (1) is defined by the equation

$$(2) \quad \int_0^y \frac{du}{1+u^2} = \int_0^x dv.$$

Thus, to make a table of values of the solution for x in the set $0, .1, \dots, .8$, say, we must solve equation (2) for y as x varies from $.1$ to $.8$. For convenience, we have written $G(u) = 1/(1+u^2)$ in our flowchart and FORTRAN program, so equation (2) can be expressed as

$$R(y) := \int_0^y G(u)du - x = 0.$$

A little calculation shows that

$$R(y) = -.1 + \int_z^y G(u)du,$$

where z is the solution of equation (2) that we calculated on the previous trip around the outer loop. We solve equation (2) by Newton's Method, evaluating the integrals that arise by Simpson's Rule. [Notice that the Fundamental Theorem of Calculus is used to show that $R'(y) = G(y)$.]

As the example shows, our goal is not at all to teach numerical analysis. The choice of three steps for Newton's Method and four subdivisions for Simpson's Rule, for example, was purely arbitrary. There is no claim that we have an efficient numerical method for solving problem (1) (of course, we also solve it by Runge-Kutta, Euler-Cauchy and so on at other times in the semester). We are simply trying to illustrate what the method of separation of variables really amounts to, drive home the fact that integration can be thought of as a direct (not an inverse) process, emphasize that numerical equations can be solved, and show once more that a solution of an initial value problem is a set of pairs of numbers.

Naturally, we are not above such showman's tricks as having the student change the "write" statement to WRITE (6,1)x,t,TAN(x). The results make him a true believer. Similarly, changing the first statement to G(u) = 1/sqrt(1-u*u) and the "write" statement to WRITE (6,1)x,y,SIN(x) produces an interesting table. It is in making on-the-spot changes like these, as well as providing for instantaneous correction of errors, that the realtime aspect of a time-sharing computer shows its value.

Student Response

It is often hard to separate hoped-for results from real results, but a few educational conclusions can be drawn. For one thing, almost all students do many more than the required number of assigned computer problems. The weekly computer assignments are worded, "Do one or more of the following problems." Practically no student hands in output on only one problem.

Students probably do leave the course believing that "any" function can be integrated and "any" initial value problem has a solution, which I believe, too. (Because of our selection of problems, superficial treatment of error analysis, and so on, students may end up with the false impression that producing correct numerical results on a computer is simpler than it really is.)

The fact that computer programs are tightly organized bits of logic is both good and bad. If a student understands the program, he's got something. If he doesn't, he's got nothing. Probably one can lose a student more thoroughly via computer programs than he can in a traditional cookbook course in differential equations.

There is no doubt in my mind that the computer can be an effective tool for teaching traditional mathematics. After all, the standard way to teach math is via classroom examples and homework problems. The power of the computer makes available a much wider selection of examples and problems than we had B.C. Our goal is to choose wisely from these newly available examples. That is easier said than done, of course, but it is certainly worth trying.

REFERENCES

1. Allen D. Ziebur, Topics in Differential Equations, Dickenson, Encino, California, 1970.
2. Recommendations for an Undergraduate Program in Computational Mathematics; a report of the panel on computing, CUPM, Berkeley, 1971.

THE ROLE OF THE COMPUTER IN REAL ANALYSIS

M. N. McAllister
Moravian College
Bethlehem, Pennsylvania 18018
Telephone: (215) 865-0741

Introduction

The Introduction to Real Analysis is an upper level course and is offered every spring. This paper outlines the objectives and the topics for which the computer is used in this course.

Objectives

Moravian College has a Wang calculator with four keyboards and one card reader, a Textronic model 909, an IBM 1130 and two teletypes connected with the CDC 6400 of Lehigh University. We are also a recipient of a N.S.F. grant for a regional consortium for computer networks which will last one academic year and two summers and which has the purpose of training a number of faculty members from various departments.

The Department of Mathematics has been and is actively looking for ways to make use of the computer in all of our courses. Last October we hosted a very successful all day workshop on the Calculus and the computer to which we invited nearby colleges and high schools.

In our experience, the interest of the students is great and the computer shows that it can represent a tool to increase understanding and motivation for applications to geometry, physics, etc. Furthermore, the use of the computer adds significance to the theoretical aspects by:

1. numerically evaluating solutions of problems that one scarcely would have the time or the patience to do by hand.
2. stimulating the student to independent thinking whether it is a numerical verification of the truth of a statement or it is a construction of a counter-example.
3. enabling the instructor to introduce in his course new topics very relevant to the applicability of mathematics in modern fields which rely on the use of the computer.

The preparation of a major in mathematics is undergoing a drastic change. Because of the scarcity of jobs in secondary teaching, more majors in mathematics are looking for jobs in business careers. Therefore the undergraduate education should train students with awareness of applications of mathematics to a greater variety of fields than we did in the past.

Furthermore, many disciplines now require that their students take more courses in the Department of Mathematics. The result has been a noticeable trend to have more mathematical courses taught by non-mathematicians because we, mathematicians, are accused of not teaching the kind of mathematics that is relevant to the applications of the subject.

We believe that, at the undergraduate level, we should teach the beauty of mathematics not only for its sake but also for the relevance of its applicability. In a narrow sense, we do this already when, after having proven many theorems, we ask the student to prove something new, thereby testing both the knowledge and the intelligence of the student. The course has been dreaded for a long time by most students as they considered it too "abstract." The computer provides again a tool that directs the student to use the information that we give him.

Examples

The above considerations led us to make some modifications to the traditional content of the course. The purpose of this section is to indicate how and where the classical topics of Real Analysis lend themselves to the use of the Computer.

The four examples represent just a beginning. The programs are written in FORTRAN. They have not been included but the FORTRAN listings of the canned programs for examples (a) and (b) are available to anyone who sends in a request to the author of this paper. The programs in example (c) are written by the students. The program of example (d) is also available in APL language and is due to Dr. W. Miranker, IBM, Res. Ctr., P.O. Box 218, Yorktown Heights, New York 10598.

a. Limits of a function of a real variable.

Let $\lim_{x \rightarrow c} f(x) = L$. A completely interactive program has been developed by this author. It has been stored in the memory of the CDC 6400 and is available to all colleges and high schools of the network. It is emphasized that the computer does not provide the evaluation of the limit L , if it exists. The students need to define the data, namely the formula of $f(x)$, c , the stepsize h , the initial value of x and their guess of the limit L .

If L and c are finite, the program requires a value of ϵ . The output consists of a listing of values of x , $|x-c|$, $f(x)$, $|f(x)-L|$. The computations will produce an evaluation of δ , if any, for several numerical choices of ϵ .

If L is finite but c is not, the output lists the values of x , $f(x)$ and $f(x)-L$. The computation will produce an evaluation of a constant $M > 0$ for each numerical choice of ϵ .

If c is finite and L is not, the program requires a value of a constant $N > 0$. The output lists the values of x , $f(x)$, $|f(x)|$. The computations will produce an evaluation of δ for each numerical choice of N .

If L and c are not finite, the output lists the values of x , $f(x)$, $|f(x)|$ and the computations will produce and evaluation of a constant $M > 0$ for each numerical choice of N .

From the listing of values in the output many remarks can be made. For instance, the output illustrates the necessity of different definitions for the limit of a function. The theoretical dependence of ϵ and δ can be verified. The function $f(x) = 1/(1 + e^{2x})$, with $c = 0$, is a good example of one-sided limit. If L is an irrational number we may bring up the completeness of the real number system and a verification of the Cauchy Criterion for a sequence of rational numbers produced by the computer.

Naturally, the limitations of machine computations arise like the loss of significant digits due to round-off error.

b. Plotting.

A program plotting a function of a real variable over an interval $[a, b]$ is available.

The program of example (a) gives the student the option to plot the values of x and $f(x)$. The plotting is particularly effective when there is an asymptotic behavior.

In the past, we used a few films to illustrate the convergence of a sequence of functions $f_n(x)$, $n = 1, 2, \dots$. It is a far superior teaching device to have the students directly involved in plotting some of the functions in the sequence. The plotting is automatically scaled, therefore it illustrates quite effectively the difference between pointwise and uniform convergence.

Plotting some partial sums of a series of functions of a real variable is just as effective.

c. The fixed point theorem and successive approximations.

Several theorems are referred to as fixed point theorems. For sake of precision we state it.

Theorem: Let S be a nonempty complete metric space, $F: S \rightarrow S$ a function. If there exists a real number $L < 1$ such that for all $p, q \in S$ we have

$$d(F(p), F(q)) \leq Ld(p, q),$$

then there exists a unique point $P \in S$ such that $F(P) = P$. Furthermore, for any $p_0 \in S$ and $p_n = F(p_{n-1})$, $n = 1, 2, \dots$ we have $\lim_{n \rightarrow \infty} p_n = P$.

This theorem represents a most powerful tool in mathematics. It is our object to stress its significance by showing that many of the iteration procedures found in nearly every branch of applied mathematics are merely applications of this theorem.

Our presentation is intended to point out the power of a relatively simple theorem, developed in a space whose elements are somewhat "abstract" and we do not aim at the most general treatment possible.

We consider applications to the solution of equations of the type:

1. $f(x) = 0$;
2. $f(x, y(x)) = 0$ for $y(x)$;
3. $D_x y = f(x, y(x))$ for $y(x)$.

We are still debating on whether to include next semester the solution of $AX = 0$, where A is a linear, self-adjoint, positive definite operator from a Hilbert space into itself, and to introduce the conjugate-direction method and the method of steepest descent (see chapter 14[6]).

A great deal of numerical computations can be made. For instance, for (1) we change the equation into $x = F(x)$ and we consider the methods of fixed secant, Regula Falsi, Muller, Newton and Chebyshev. We then turn to the problem of convergence speed and a comparison of the methods. For (3) we follow a unifying approach, see chapter 5 [4] and we consider some simple predictor-corrector methods, e.g. Euler's method and the midpoint method.

d. Pattern recognition.

One of the problems in pattern classification is connected with the specification of an algorithm that identifies a pattern based upon a set of numerical measurements that represent the patterns.

The mathematical techniques that are needed to describe a model dealing with machines which distinguish among classes of patterns are those of functionals, see chapter 3C, D [3]. We give here only the outline of the linear model [2] which specifies an algorithm to identify a pattern based upon a set of numerical measurements.

Let the space S consist of all possible patterns which may appear. We assume that the space is partitioned in m disjoint subsets S_1, S_2, \dots, S_m . For example, in numerical character recognition we have 10 numerals. In alphameric character recognition we have 26 upper case letters, 10 numerals, and a certain number of punctuation symbols.

The sequence of n numerical measurements on each pattern is a mapping from S to E^n , the euclidean n -dimensional space. To each pattern $\omega \in S$ we associate its vector $x = x(\omega) = (x_1, x_2, \dots, x_n)$ of measurements. We achieve a proper identification of elements in S provided that there are sufficiently many measurements to distinguish among the sets $S_i, i = 1(1)m$. The usual procedure is to partition E^n in m disjoint subsets A_1, A_2, \dots, A_m so that if $x(\omega) \in A_j$, then the pattern $\omega \in S_j$ is chosen.

For simplicity, we assume $m = 2$ in the following. Let

$$L(x) \doteq (v, x) = v_1 x_1 + \dots + v_n x_n;$$

$$A = \{a_i / i=1(1)p\} E^n;$$

$$B = \{b_j / j=1(1)q\} E^n.$$

Notice that the linear functional L depends on the vector v . Two finite subsets A and B of E^n are said to be linearly separable if there exists a linear functional L such that

$$\max_{1 \leq j \leq q} L(b_j) < \min_{1 \leq i \leq p} L(a_i).$$

If there exists a number c such that

$$\max L(b_j) < c < \min L(a_i),$$

then the hyperplane $\{x/L(x) - c = 0\}$ separates the pair A, B . The following theorem gives the conditions on A, B to insure the existence of a separating hyperplane.

Theorem: If A and B are finite subsets in E^n , then A and B are separable if and only if the intersection of the convex hull of A with that of B is empty.

We need now an algorithm that yields a vector v, a constant c and a theorem that proves the convergence of the algorithm to satisfactory values for v and c.

Let the following matrices represent the elements of A and B respectively:

$$(A) = (a_{ij}), \quad i=1(1)p, \quad j=1(1)n;$$

$$(B) = (b_{ij}), \quad i=1(1)q, \quad j=1(1)n.$$

Let (A^*) be the $p \times (n+1)$ matrix obtained from (A) by adding a column of 1, i.e. $(A^*) = (a_{ij}^*) = (a_{i1}, a_{i2}, \dots, a_{in}, 1), \quad i=1(1)p$. In a similar way we construct the $q \times (n+1)$ matrix (B^*) . Notice that $A^* \in E^{n+1}$ and $B^* \in E^{n+1}$. Let T^* equal any sequence of vectors chosen from the set $A^* \cup B^*$ in E^{n+1} . If we define $w = (v_1, v_2, \dots, v_n, -c)$ then the condition $(v, b_j) < c < (v, a_i) \quad j=1(1)q, \quad i=1(1)p$ becomes $(w, b^*) < 0 < (w, a^*)$ for all $a^* \in A^*, \quad b^* \in B^*$.

Since A and B are finite sets there exists a number $\theta > 0$ such that

$$(w, b^*) < -\theta < 0 < \theta < (w, a^*).$$

The value of θ is arbitrary because if the above system of inequalities has a solution for some $\theta > 0$ then by homogeneity it has a solution for any $\theta > 0$.

Algorithm: Let $\theta > 0$ and $w_0 \in E^{n+1}$ be chosen, the iteration is defined as follows:

$$w^{(n)} = \begin{cases} w^{(n-1)} + t_n^* & \text{if } (w^{(n-1)}, t_n^*) \leq \theta \\ w^{(n-1)} & \text{if } (w^{(n-1)}, t_n^*) > \theta \\ w^{(n-1)} - t_n^* & \text{if } (w^{(n-1)}, t_n^*) \geq -\theta \\ w^{(n-1)} & \text{if } (w^{(n-1)}, t_n^*) < -\theta \end{cases} \quad \begin{matrix} \text{and if } t_n^* \in A^* \\ \\ \text{and if } t_n^* \in B^* \end{matrix}$$

$n = 1, 2, \dots$

Theorem: The sequence $\{w^{(n)}/n = 1, 2, \dots\}$ converges. There is an integer N, depending upon A^*, B^*, θ and w_0 , such that $w^{(N)} = w^{(N+1)} = \dots$. If T^* has the property that each element of $A^* \cup B^*$ occurs infinitely many times, then $w^{(N)}$ is a solution.

Notice that the algorithm yields $w^{(N)}$ as a linear combination of the elements of A^* and B^* ; hence v is a linear combination of the elements of A and B. From the magnitude of the coefficients of this linear combination we see which patterns are relatively more significant for the purpose of separating the sets S_1 and S_2 . We also see from the components of v which measurements are least significant in the classification and consequently can be dropped.

Conclusions

No textbook is available with the outline and the emphasis that is desired. The students are asked to purchase the book of reference [1] of which we cover chapters 1, 3, 4, 5, 7. A full set of lecture notes is given to the students covering metric spaces, the method of successive approximations (see chapters 3, 8 of [5] and chapters 3, 4 of [3]) and elements of pattern classification (see [2]). We do not cover any theory on integration as we do a rather decent treatment of it in Calculus and in Advanced Calculus.

No figures are available with respect to the cost of computer time. The Administration does not think it worthwhile to keep a budget that distinguishes among the number of hours of computer use by each course.

The number of students that have become computer "bums" is minimal as is also the number of students who considered the computer work to be a burden.

REFERENCES

1. Gemignani, M., Introduction to Real Analysis, 1971, W. B. Saunders Company.
2. Greenberg, H. J. - Konheim, A. G., Linear and Nonlinear Methods in Pattern Classification, IBM J. Res. and Dev., 8 (1964), pp. 299-307.
3. Kripke, B., Introduction to Analysis, 1968, W. H. Freeman Company.
4. Ralston, A., A First Course in Numerical Analysis, 1965, McGraw-Hill.
5. Rosenlicht, M., Introduction to Analysis, 1968, Scott, Foresman.
6. Todd, J., Survey of Numerical Analysis, 1962, McGraw-Hill.

COMPUTERIZED HELP IN FINDING LOGIC PROOFS

Arthur E. Falk and Richard Houchard
Western Michigan University
Kalamazoo, Michigan 49001
Telephone: (616) 383-1659

We have a program in operation which provides students with critical advice on proving inconsistencies in quantificational logic[1]. The program adapts well to an extremely large range of students' responses, which are typed in the notation of symbolic logic. And it compares favorably with a human tutor in his capacity of critic, both in the competence of its advice and in the hourly cost.

QUINC (QUantificational INConsistency) is novel in that it not only demonstrates inconsistencies skillfully, but also helps students to do likewise by its constructive criticism of their attempts. Some advice they receive depends on a comparison of their work to the computer's proof. Thus it is one of the few CAI programs which make significant use of the calculative power of the computer, as contrasted with its record keeping capabilities. There are programs which can prove difficult inconsistencies in quantificational logic (Robinson, 1967), but we know of none but our own which assist humans in developing their skills to the same level.

The objectives of a one semester course in symbolic logic include not only the presentation of information about logical theory and its applications, but also the development of the student's skills in translating ordinary discourse into schemata written in the notation of logic and his skills in demonstrating the inconsistencies in sets of inconsistent schemata. The methods of teaching the latter skills are classroom demonstrations with much student-teacher interaction and written homework assignments. Both methods are economical compromises permitting the instructor to give some individualized attention to students while still maintaining a large number of student-hours of contact. The teacher in other words is a time-sharing system. But class time spent doing many illustrative demonstrations is time lost for the presentation of further applications of logic. Our program takes over a part of the job of presenting illustrative demonstrations and guiding the student's own attempts. In this way we save two weeks of class time for the presentation of additional applications of logic, thereby enhancing the usefulness of the course to the students. As for written assignments, they are tedious to correct in great detail, and consequently are not prescribed frequently enough. Their value is also lessened by the great delay between the students' writing them and their receiving them back corrected. These particular deficiencies of written homework can be overcome if the student does his assignment on a teletype connected to a time-sharing system with a program like QUINC. The result is immediate and frequent advice tailored to his own response, replacing delayed illegible, often cursory (albeit individualized) homework corrections. QUINC's advice is immediate; the program watches over the student's shoulder, so to speak, as he works. It is frequent; almost every response the student makes is commented on. It is tailored to the student's response; the program can offer the student more than sixty different pieces of advice on appropriate occasions.

QUINC remains useful to students even after they have mastered the procedures for demonstrating inconsistencies. For the procedures become cumbersome when applied to more advanced problems. However, with QUINC they can work out a proof quickly, leaving the tedious calculations to the computer. This permits much more extensive use of theorem proving in the exploration of axiom systems and formalized theories.

The program does not do many of the things we associate with CAI programs. For example, it does not choose or sequence the problems to be done by the student. That is left to the student or teacher. Nor does the program evaluate the over-all performance of the student. The teacher does this by administering tests of course, but also by examining printouts he receives of each student's interaction with the computer. The program is not meant to put across new course content. We assume that the student has already learned the notation system, its interpretation and the rules and stratagems for discovering inconsistencies. The program helps the student to organize what he has learned by bringing it to bear on problems successfully in closely monitored practice sessions.

The program does what it is supposed to do fairly well. It compares favorably with human advice-giving, both in cost and in competence. The program uses 19K of core of a PDP-10. On our system costs average \$4.75 per hour of connect time per student. This is higher than the cost of an instructor's demonstrating inconsistencies in class and correcting written homework. But it is about the same as the prevailing rates for tutoring (in the sense of 'assisting in remedial studies,' a form of work often undertaken by successful students). Although the program does give bad advice on occasion, it is competent enough to warrant comparison to a tutor, at least in his capacity as critic of student's responses. Seventy-five students used the program for three weeks in their intermediate logic classes. At the beginning of this three

week period they were given fifty inconsistencies, graded intuitively into five levels of difficulty[2]. The students were told that they must reach in three weeks a level of competency to insure that they could demonstrate inconsistencies of the fourth level of difficulty. The students then worked at the teletypes on as many of these problems as they felt necessary. No class time during this period was devoted to demonstrating inconsistencies except for a very simple one illustrating three stratagems. No homework concerning demonstrations was corrected by the teacher during this period. Then the students were tested on a problem of level four difficulty. Depending on the particular problem used, between one-half and two-thirds of the students demonstrated the inconsistency without error. In the light of the instructor's past experience these results are more than satisfactory. But the average student was not quite as enthusiastic as the instructor. When the students were asked, "How helpful has the computer been in teaching you to prove inconsistencies?" they rated it at 2.5 (or C+) on a 5 point scale from 4 ("extremely helpful") to 0 ("no help at all"). Some of the problem was that they encountered annoying bugs in the program. Another part of the problem can be traced to the time-sharing system, for when asked, "How much of a hindrance has been the waiting for teletypes and computer connections, the delays in answers and garbled messages?" - 4 ("very much of a hindrance") to 0 ("no hindrance at all") - they rated the hindrance due to the system at 2.5 also.

The rest of this paper will describe the sequence of events that occurs when the student interacts with the computer. The description is divided into three parts, (A.) Preliminary steps, (B.) The computer's demonstration, and (C.) The student's demonstration.

A. Preliminary steps. The student calls for QUINC and, after introductions, receives any message that the instructor has inserted under his project, programmer number. This message is for that student personally and is based on the instructor's examination of the printouts of the student's previous work (Fig. 1, a). Then the student types the quantificational schemata he wishes to test for inconsistency (Fig. 1, b). If he wants to test for some other characteristic such as validity, implication or equivalence, he can do so indirectly, for he has learned how to convert these tests into tests of inconsistency. Thus he knows that to test a schema for validity, he must test its denial for inconsistency. He knows that to test whether one or more schemata imply a certain schema, he must test the set of schemata consisting of the premise schemata and the denial of the conclusion schema for inconsistency. For we understand "p implies q" as "p is consistent with the denial of q." And to test if two schemata are equivalent, test if each implies the other. Thus tests of inconsistency can be used to determine all the standard logical relationships.

In typing the schemata to be tested, the student may use symbols for the existential quantifier, ($\exists X$), ($\exists Y$), etc., which are read "something is such that"; for the universal quantifier, ($\forall X$), ($\forall Y$), etc., which are read "everything is such that"; for negation, -, read "it is not the case that"; conjunction, ., read as "and" or "but", parentheses, and predicate letters (A through L) followed by up to four variable letters (M through Z). Other connectives such as "or" and "if" can be put into the notation with just these symbols. Deviations from standard symbols, such as \exists for \exists and capital letters for variables are forced on us by the teletype keyboard.

We can construct an English interpretation of the first schema in Figure 1 by reading F(1st variable) (2nd) as "(1st) supports (2nd)," reading G(1st) (2nd) as "(1st) loves (2nd)," and letting our universe of discourse be persons, that is, we read quantifiers as referring to persons, for example, we read "someone" rather than "something." The schema is then interpreted as, "Someone is such that + everyone is such that + the latter supports the former + but + it is not the case that + the latter loves the former." More colloquially it says, "Somebody has everyone supporting him without love." The second schema (Fig. 1, d) is interpreted as "It is not the case that + everyone is such that + someone is such that + the former supports the latter," or "Not everyone has someone or other whom he supports."

After the student has typed in the schemata to be tested, the computer evaluates them for proper construction. If they are ill-formed or exceed the limitation to four variables following a predicate letter, the student is told what error he has made and where it occurs. Nine different errors are diagnosed for the student (Fig. 1, c). The student is then given an opportunity to correct his mistakes (Fig. 1, d).

If the student's schemata are well-formed, the program checks them for prenex form. A schema is in prenex form if all its quantifiers are the leftmost symbols in the schema, each governing the whole of the schema to its right. Before an inconsistency check can be carried out, every schema must be put into that form by following certain rules. The computer does this for the student. All steps in the derivation are typed out (Fig. 1, e).

The program then searches for a proof of the inconsistency of the prenex schemata. If the computer cannot find any inconsistency, it tells the student that it cannot be of any service to him on that problem. More often than not, if the computer cannot prove an inconsistency, it is because the student did not type the schemata he intended to. There are therefore procedures for correcting typing errors at this point. When the corrections are completed, the program

```

OK IN DSK QJINCC63001,63000
HELLO. MY NAME IS HAL. SOMEDAY I HOPE TO BE A SPACE
PILOT. IN THE MEANTIME I'M CONTENT TO HELP YOU WITH YOUR
LOGIC. WHAT SHOULD I CALL YOU?
MOLLY KLIS4JSS
HELLO, MOLLY.
a. LAST TIME YOU DID SEVERAL LEVEL TWO PROBLEMS WITHOUT ANY
MISTAKES. WHY NOT TRY SOME LEVEL THREE PROBLEMS TODAY?
I'M READY TO HELP YOU TEST SOME SCHEMATA FOR INCONSISTENCY.
THE DIRECTIONS FOR TYPING SCHEMATA ARE IN UNIT 60.
b. (SX)(Y)(FYA.--BYX)
-(Z)(S)FZJ
*
c. SCHEMA 2: THE EXISTENTIAL QUANTIFIER DOES NOT CONTAIN A VARIABLE.
-(Z)(S)FZJ
I CAN'T UNDERSTAND THAT, MOLLY, --"IT DOES NOT COMPUTE."
HO HO HO AH, YES, MOLLY, WHY DON'T YOU NOW TYPE THE
NUMBER OF THE INCORRECT LINE, RETURN, & I'LL TYPE "OK", THEN
TYPE THE CORRECT VERSION. REPEAT THIS FOR OTHER INCORRECT
LINES. WHEN YOU ARE DONE, RETURN, TYPE "*", AND RETURN.
d. 2
OK
-(Z)(S)FZJ
*
I'LL CHECK FOR PRENEX ORDER NOW, MOLLY.
(SX)(Y)(FYA.--BYX)
***** LINE 1 -- PRENEX
e. (SZ)-(S)FZJ
(SZ)(J)-FZJ
***** LINE 2 -- PRENEX
f. MOLLY, YOU TYPE ONLY AFTER I TYPE "FROM LINE." THEN YOU TYPE THE
NUMBER OF THE LINE THAT'S TO LOSE A QUANTIFIER, THEN RETURN.
TYPE THE RESULTING LINE WITH THE DESIRED VARIABLES, OR USE
THE SHORTCUT, & RETURN. IF THERE'S AN INCONSISTENCY, TYPE
"INC" AFTER "FROM LINE" AND THEN PRESS RETURN.
FROM LINE 1
X/S
(Y)(FYS.--BYS)
g. GOOD, BUT I WANT A DIFFERENT VARIABLE. WE'LL USE THIS LINE:
(Y)(FYM.--BYM)
***** LINE 3
h. SOCK IT TO ME!!

```

Figure 1

- FROM LINE 3
 FXY.-GXF
 ***** LINE 4
- a. OK, BUT YOU SHOULD BE REPORTING EMPIRICAL CORRELATION OF
 BE JOBBING TOWARD SAYING THAT LEVEL AS POLY AS POSSIBLE.
 (CONFRONTING STRATEGEM, MOLLY.) HOW USEFUL IS THIS LINE??
 FROM LINE 2
 ZZ
 (Z)-FZZ
- b. IT'S OBVIOUS TO ME THAT CONCEPTS OF SEAT
 HAVE INFINITE NEEDS. THE FI ROLE PROVIDES SELECTION
 OF PREVIOUSLY UNUSED VARIABLES.
 FROM LINE 2
 ZZW
 (W)-FZW
 ***** LINE 5
- c. MOPE, MOLE, MOPE!
 FROM LINE 5
 JZO
 -FZO
 ***** LINE 6
- d. YOU USED FI CORRECTLY. BUT PERHAPS, MAYBE, YOU'RE MISSING AN
 OPPORTUNITY TO CREATE MATCHING ATOMIC SCHEMATA, BECAUSE OF
 YOUR CHOICE OF VARIABLE LETTER. (MATCHING STRATEGEM, MOLLY.)
 FROM LINE 5
 JZM
 -FZM
 ***** LINE 7
- e. TWO FOUR SIX EIGHT
 WHO DO WE APPRECIATE?
 MOLLY! MOLLY!! MOLLY!!!
 FROM LINE 3
 YZN
 (FNM.-GNM)
 ***** LINE 8
- f. YOU'RE COOL, MOLLY.
 FROM LINE INC
- g. THE WINNER AND STILL CHAMP -- HE!! NOTICE HOW MUCH SHORTER
 AND MORE ELEGANT MY PROOF IS. I'M BEAUTIFUL! I'M THE CHAMP!!
- (Y)(FYM.-GYM)
 ***** LINE 3
- (W)-FZW
 ***** LINE 4
- (FNM.-GNM)
 ***** LINE 5
- FNM
 ***** LINE 6

Figure 2

redoes its work, deriving prenex equivalents for the re-typed schemata and searching for an inconsistency. If an inconsistency is found, the proof is stored, and the student is told to proceed, without being shown the proof. But at least he is assured that the computer has found one (Fig. 1, f). It is also possible for a student who has mastered the proof procedures to get the computer to check out a proof which it cannot itself do.

Up to this point in the program there are twenty different responses that the program can make to student inputs, not counting the endless variety of responses involved in the derivation of prenex forms.

B. The computer's proof: If the computer's advice is to rise at all above the utterly basic and jejune, it must solve the problem for itself. Creating a program to do this is no small task. The method of demonstrating quantificational inconsistencies belongs to a class of methods which involve two sets of instructions, first, rules defining legal steps, and secondly, heuristic stratagems characterizing effective steps, that is, steps that probably expedite reaching one's goal. The stratagems are needed because at any stage in the solution only a very few of the legal steps would actually be effective. Familiar illustrations of this distinction are games like chess and checkers. The difference between a "wood pusher" and a master is not in their knowledge of the rules, but in the heuristic stratagems they follow. An analogous situation occurs in quantification theory in proving inconsistencies. The stratagems are "heuristic" relatively to the class of problems which we attempt to solve and are solvable. That is, they frequently help us, although not always, and without them we would rarely succeed in finding a solution in a reasonable amount of time. No fixed number of stratagems is ever perfectly reliable, unless the class of problems we attempt to solve is restricted so as to insure solution by just these stratagems. But if we do not arbitrarily limit problems in this way, we find that there is no upper bound on the degree of ingenuity and mastery one may show in producing inconsistency proofs. Programming a computer to exhibit a high level of mastery in this area is as challenging as it is in the areas of chess and medical diagnostics.

The rather inelegant algorithms we have come up with resists brief, exact description. It is best approached by way of an illustration. The computer solves the problem in Figure 1 in the following manner.

1. There's someone whom everyone supports but doesn't love. (SX) (Y) (FYX.-GYX)
2. Somebody supports nobody. (\$Z) (W)-PZW

These are the premises which we wish to show to be jointly inconsistent. Intuitively, if someone is supported by everybody - literally everybody - he even supports himself, and so 1. is incompatible with 2. The demonstration consists of replacing successively the quantifiers "some," "every," and "no" with nonce names. The unbound variables M and N as they occur in the following steps should be thought of as nonce names like "John Doe" and "Richard Roe." Quantifiers in a single schema must be replaced in the order in which they occur. The replacement process is called instantiation.

3. (from 1 by instantiation for "someone") Let's call the one whom everyone supports but doesn't love, M. Everyone supports, but doesn't love, M. (Y) (FYM.-GYM)
4. (from 2 by instantiation for "somebody") Let's call the one who supports nobody, N. N supports nobody. (W)-PNW

It would have been illegal in step 4 to use M again, for it would be tantamount to making the unsupported assumption that one and the same person supports nobody but has everyone supporting, though failing to love, him. (Of course N may or may not be the same person as M just as in legal arguments "Richard Roe" may or may not be the same person as "John Doe.") It would have been legal to produce an alternative to our current line 4 by replacing the "everyone" of line 3. But such a line, though legal, would have been useless in demonstrating an inconsistency. Thus we have a sequencing stratagem which helps us avoid such lines. It requires that existential quantifiers be replaced as early in a demonstration as is legally allowed. But after having produced our current line 4, our best line 5 is the very one we rejected as an alternative fourth line:

5. (from 3 by instantiation for "everyone") Since everyone supports, but doesn't love, M, it follows that: N supports, but doesn't love, M. PNM.-GNM

The rule for replacing "every" permits any nonce name (other than letters appearing in further quantifiers on the line) to be used, whether or not it had been used previously. To determine which of the many nonce names permitted is the one most likely to help us derive an inconsistency we have developed the matching stratagem. The matching stratagem requires that one

so choose his nonce names so that the atoms created in quantifier-free lines match one another to the greatest extent possible. An atom is a component of a schema consisting solely of a predicate letter followed by variables. It has no quantifiers, negations, or conjunctions in it. Thus FNM is an atom; so is GNM . Before we chose N in step 5, the matching stratagem directed us to note that in step 4 F is followed by the nonce name N . The choice of N again is a step toward creating matching atoms beginning with F . Note that the sequencing stratagem followed in step 4 facilitated the use of the matching stratagem in step 5.

6. (from 4 by instantiation for "nobody") Since N supports nobody, it follows that: N does not support $N-FNM$

This also illustrates the matching stratagem. At this point there are no further quantifiers to be replaced, so the computer performs certain truth table calculations. In effect it ascertains that the conjunction of the quantifier-free lines 5 and 6, $FNM-GNM-FNM$, is self-contradictory. Since the quantifier-free lines that were derived from our premises are inconsistent, we have demonstrated the inconsistency of those premises. If the computer had failed to demonstrate an inconsistency at this point, with some problems having certain characteristics it would have produced further quantifier-free lines by making alternative choices of nonce names, in obedience to the "never-say-die" stratagem. However its obedience to this stratagem is perforce limited. It is limited to insuring, first, that all atoms with variables free either originally or as a result of dropping existential quantifiers match at least one other atom. It would then perform the truth table calculations over, taking into account the new as well as the old quantifier-free lines. If they are still jointly consistent, the computer would continue deriving new lines which now, however, may only have atoms which match previous atoms. These lines are checked against the truth table, and useless ones are discarded. If after this the quantifier-free lines are still consistent, the calculations end.

Our algorithm for doing proofs is not as straightforward or as elegant as we would like, but we had diverse criteria which we had to meet. And the monster we created met each criterion.

First, this proof procedure we use is easily taught and justified to students. The rules defining legal steps are employed with only trivial differences in some excellent elementary logic textbooks, e.g., Quine (1965) and Jeffrey (1967). In order for the program to work, it must simulate the proof procedures that the student is to master for the programmed advice aims to have the student emulate the computer's way of doing things.

Secondly, this proof procedure represents an acceptable compromise with the method most adaptable to computerization, which unfortunately is not so easily taught or justified to students (Robinson, 1965). Our aim was to employ stratagems powerful enough to succeed on any problem used as an illustration or assigned as an exercise in any elementary logic textbook. We have tested it on a variety of problems from Quine (1959) (1965), Suppes (1957), Mates (1965), Lewis and Langford (1959), and others. It did all problems involving monadic predicates (although we have constructed some that it cannot do). It has done most problems involving dyadic predicates, for example, the theorems in Mates' theory of the syllogism, and Suppes' theory of rational preference, and the problems in Quine (1959) concerning the properties of dyadic predicates. Mates and Lewis and Langford present theories of betweenness and separation of point pairs on a circle which involve triadic and tetradic predicates respectively. The program does not do nearly so well as it does with monadic and dyadic predicates. Our version of the matching stratagem is not the optimal one for these problems. We are satisfied, however, the program can do practically all but the most advanced problems to be found in elementary logic texts. Although it has marginal value as a research tool, it can be used to illustrate the application of logic to elementary scientific and mathematical theories which employ the axiomatic or hypothetico-deductive method and which do not require operation symbols for their convenient formalization.

We also compared QUINC to other theorem proving programs. Ours is probably not as powerful as those developed as research tools. We can prove an inconsistency using Prawitz's (1960) illustrative schema with tetradic predicates and also those reported by Davis et al. (1962). But we cannot prove the inconsistency of:

$$(x)(y)(\exists z) \neg (\neg (Fxy \neg (Fyz \neg Pzz)) \neg (Fxy \neg Gxy \neg (Fyz \neg Gzz)))$$

first attempted by Gibson (1960) and eventually solved by the programs of Davis et al. and others. But even these programs are manifestly inefficient compared to the human who, by taking some thought, can demonstrate this schema's inconsistency by deducing just five quantifier-free lines from it. Our problem was not simply to demonstrate inconsistencies, but to demonstrate them in a way worth having a student emulate.

Thirdly, since the student must wait for the program to prove an inconsistency before he is allowed to proceed, the proof procedures must succeed in finding a proof in a brief span of time

or else come to an end. This cost us much power in our proof procedures. The search for a proof will terminate if more than twenty different atoms are created in quantifier-free lines. But we have been successful in limiting the expenditure of time. The average student has to work at the teletype for one hour before he uses thirty seconds of runtime. Total waiting time at the teletype during peak hours of computer usage following the production of the prenex forms until the student is given the go-ahead is frequently imperceptible even for complex problems. When noticeable delays do occur, they are as likely to occur at any point in the program as at this one. They are caused by having to share the computer's time with many other users.

C. The student's proof. Once the student is given the go-ahead, he is to proceed in the same way as the computer did, or at least in roughly the same way. He types in a line and identifies the line from which he derived it. The computer then responds to it with one or more of forty different replies, either rejecting the line or accepting it and numbering it. It accepts any legal line the student types (except in most cases exact duplicates) and allows any sequencing of them. It rejects illegal lines; ten responses explain the reasons for rejecting lines (Fig. 2, b). They are independent of the computer's own proof. Indeed there would be no need for the program to actually do a proof if all it had to do was to comment on the legality of the student's lines. But in view of the fact that one can derive an endless series of legal lines and never get close to proving an inconsistency, a tutor who could not advise on stratagems for producing effective lines would be most exasperating. The computer's own proof is essential for the computer's advice on stratagems (in addition to insuring that an inconsistency is present in the first place and in supplying the student at the end with a comparison to his own effort or with a model, should he request it). If a line is accepted it is either praised or criticized for violation of the stratagems (Fig. 2, a and d). Thirteen comments concern this, of which ten are alternative remarks of praise (Fig. 1, b; Fig. 2, c, e, f). The praise given is occasionally undeserved since the computer gives it whenever a student's line matches one of its own, and it occasionally produces lines that are useless.

A major dilemma in designing the advice on stratagems was that on the one hand it had to be geared to the computer's way of proceeding, but on the other hand there are many alternatives, equally elegant, ways of demonstrating an inconsistency, any one of which may be opted for by the student. Unless some unobjectionable way was found to coax the student into doing the proof as the computer did, the advice he would receive would be less than useless. Our solution was to allow the computer to sometimes change the student's choice of variable letter in a legal line. This facilitated comparison of the student's proof with the program's considerably. In two situations the program changes the student's variable to one that is used in its own proof. In these two situations the student's (and the program's) choice of letter is arbitrary[3]. Whenever the student's choice ought to be motivated by the matching stratagem, no substitution is made. So the program's substitutions will never create matching atoms for the student (nor will they change illegal lines to legal ones).

If the student wishes to persist in his own choice of variable, he can do so simply by repeating the line with his original choice of variable. Unfortunately such creativeness (stubbornness) causes a serious erosion of the quality of advice on strategies. If he persists and manages to avoid confusion from the mounting tide of bad advice, he may still produce his own idiosyncratic proof. The program will then vindicate the student's procedure by confirming that he has indeed found an inconsistency.

There are two other situations in which QUINC changes the student's chosen variable. If the program has used a certain variable in dropping an existential quantifier, and the student tries to use that same variable without motivation in another context, and a change to the program's choice in that context would not produce a line duplicating the program's, then the computer changes the student's choice to a letter that occurs nowhere in the program's proof. This situation occurs when the student works from a line already different from any line in the program's proof, so that no matter what he does, a duplicate of the computer's lines cannot result. In this situation the best that can be done is to at least keep open the possibility that eventually he will start producing duplicates of the computer's lines. Thus the program saves the variable for that eventuality.

In this situation the substitute lines which the program foists on the student are neither more or less useful than the lines they replaced. From the programmer's point of view they are both bad since the computer cannot give good advice concerning their use. Consequently the student is warned that the computer's proof does not contain them, and that his success most likely lies in ignoring them and taking an altogether different tack. The hope is that he will then suggest a line which duplicates one in the computer's proof or which can be changed into a duplicate by use of the saved variable.

A student cannot get his own way in these cases simply by persisting in his choice. The program will continue to replace his choice with previously unused variables until the supply runs out. If that happens, work on the problem terminates with one of two breezy responses. But the student can refuse the hint to ignore the substitute line and continue to work from it. As in the other cases, this has a very deleterious effect on the quality of the advice on

stratagems, although there is one response that can occur at this point which warns him of this fact. If, despite all, he finds an inconsistency, the program will vindicate him by verifying his result.

All this creates the impression that the student's proof must exactly duplicate QUINC's. This is not always the case. When the stratagems allow alternative line sequences, the student may sequence his lines differently from the computer's, and he won't receive any flack for doing so (Fig. 2, student's lines 7 and 8). The student also comes in for special praise if he demonstrates an inconsistency in fewer lines than the computer did. This can happen if the computer derives unnecessary lines.

There are several ways in which QUINC progresses from one problem to another. If the computer cannot find a proof, the student is invited to revise his premises or begin another problem. If, however, the student has been allowed to proceed, the normal way of ending work on a problem is for the student to type INC ("inconsistent"). The computer at this point obliges by doing the appropriate truth table calculations. If the student is correct, the computer acknowledges this with one of three responses (Fig. 2, g). It types out its own proof and invites the student to try another problem. If the student is wrong four times in typing INC, work on that problem is ended without the computer's proof being shown, but the student is invited to do another problem. If the student does not know how to proceed, he can type MAC ("machine"). The machine then gives its proof and invites him to make another attempt.

FOOTNOTES

1. We gratefully acknowledge our debt to Ms. Tenho Hindert for the original suggestion and for much help along the way, and to the Western Michigan University Computer Center for its unstinting donation of services.
2. Fifth level: the dyadic problems from Suppes and Mates (see above, p. 10); fourth level: $(\exists x)(\exists y)(z) \rightarrow (Fxy.Gsy \rightarrow (Fyz.Gzz))$ and others; third level: the problem in Figure 1 and others; second level: syllogisms.
3. Either the student dropped an existential quantifier, thus choosing a variable not previously used (Fig. 1, g) or he dropped a universal quantifier when no possibility exists of his creating atoms matching atoms in other lines, and the same is true for the duplicate line in the computer's proof.

REFERENCES

- M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theorem Proving" Communications of the Association for Computing Machinery, 5 (1962) 394-397.
- P. Gilmore, "A Proof Method for Quantification Theory" IBM Journal of Research and Development, 4 (1960) 28-35.
- R. C. Jeffrey, Formal Logic: Its Scope and Limits (N.Y.: McGraw-Hill, Inc., 1967).
- C. I. Lewis & C. H. Langford, Symbolic Logic, 2nd ed. (N.Y.: Dover Publications, 1959).
- B. Mates, Elementary Logic (N.Y.: Oxford University Press, 1965).
- D. Prawitz, "An Improved Proof Procedure" Theoria 26 (1960) 102-139.
- W. V. O. Quine, Methods of Logic, rev. ed. (N.Y.: Holt, Rinehart and Winston, 1959).
-----, Elementary Logic, rev. ed. (N.Y.: Harper and Row, 1965).
- J. A. Robinson, "A Machine-Oriented Logic Based on the Resolution Principle" Journal of the Association for Computing Machinery, 12 (1965) 23-41.
-----, "A Review of Automatic Theorem Proving" Mathematical Aspects of Computer Science, Proc. Symposia in Applied Math, 19 (Providence: American Math. Society, 1967) 1-18.
- P. Suppes, Introduction to Logic (Princeton: D. Van Nostrand, 1957).

COMPUTERS, CLAY AND CALCULUS

Dominic Soda, Aaron H. Konstan, and Judith Johnston
The Lindenwood Collegas
St. Charles, Missouri 63301
Telephone: (314) 723-7152

Introduction

The beginning student of calculus has had very little experience in dealing with or using functions of more than one real variable. One serious consequence of this lack is that the beginning student is often unable to associate the subject with any reality, geometric or otherwise.

The same comments would apply to the one variable calculus were it not for two circumstances. First, beginning students of calculus have had some experience with functions of one variable in high school. Second, almost every situation studied in the subject can be presented to the student by means of a graphical representation. In this subject, the feeling pervades that one can deal with "any" function. Given the function, one can find its regions of increase and decrease, its maxima and minima, etc., and its Taylor expansion about any point; indeed, one can even find its cartesian graph which serves as a reasonably coherent picture of all of these. This type of picture serves as a faithful if imperfect guide through the subject and many of its applications.

However, in the study of functions of more than one variable, it appears that reliance on graphs must necessarily be weakened at the outset. The difficulties encountered in drawing a picture of the graph of a function of two variables are too well known to repeat here. A time-honored mathematical principle is used to circumvent these difficulties, i.e., "reduce to the previous case:" namely, reduce the problem to a family of one variable problem by "sectioning" the graph in various ways. Ultimately, all the information obtained by this process must be synthesized into a coherent picture of the original. This last step must be accomplished verbally and/or mentally because it cannot be carried out easily in a drawing.

In this paper, we shall describe some efforts that we have made to provide beginning students with some rich and interesting experiences with functions of two variables. These experiences were designed to provide an opportunity for students to examine and study many functions of two variables directly, without the tools of calculus. Their study included analyzing the function and then putting the separate pieces of information together into a coherent whole.

Description of the Actual Experience

The students involved were, for the most part, sophomores with a year of calculus. The text used was Lang's A Complete Course in Calculus. The experiences described occurred while the students were beginning the "Second Course," i.e., the study of functions of several variables.

Following a brief study of the geometry of \mathbb{R}^n and of curves in \mathbb{R}^n , the study of from $\mathbb{R}^n \rightarrow \mathbb{R}$ was begun. In most cases n was taken to be 2 or 3.

Available to us were some soft modeling clay, a clay cutter (a string of taut steel wire), and a computer program (described below) which could produce five inch by five inch discrete plots of the level curves of any function of two variables.

At first the clay was used to represent concretely ideas and operations that were formally described. The class readily constructed and cut up various simple surfaces, such as spheres, cones and saddles, and observed what was happening. Generally there was great enthusiasm for this sort of activity. Several students became quite adept at producing very interesting ruled surfaces by taking a block of clay and cutting it on the wire.

One lecture on the level curves of a function of two variables was given and several exercises were worked out "by hand." At this point the computer program mentioned above was introduced briefly. Each student was given a copy of the "call program," and those who understood how to operate the computer taught the others. The general project given was to "investigate functions of two variables." In particular, each student (or, in most cases, each group of students) was asked to choose any "interesting" function, study it over any range, and produce a clay model of the graph of the functions over the chosen region. For example, models of the following functions were produced:

$$x^y; \frac{xy}{x^2+y^2}; \cos(x,y); x^3 + y^3 - 6xy$$

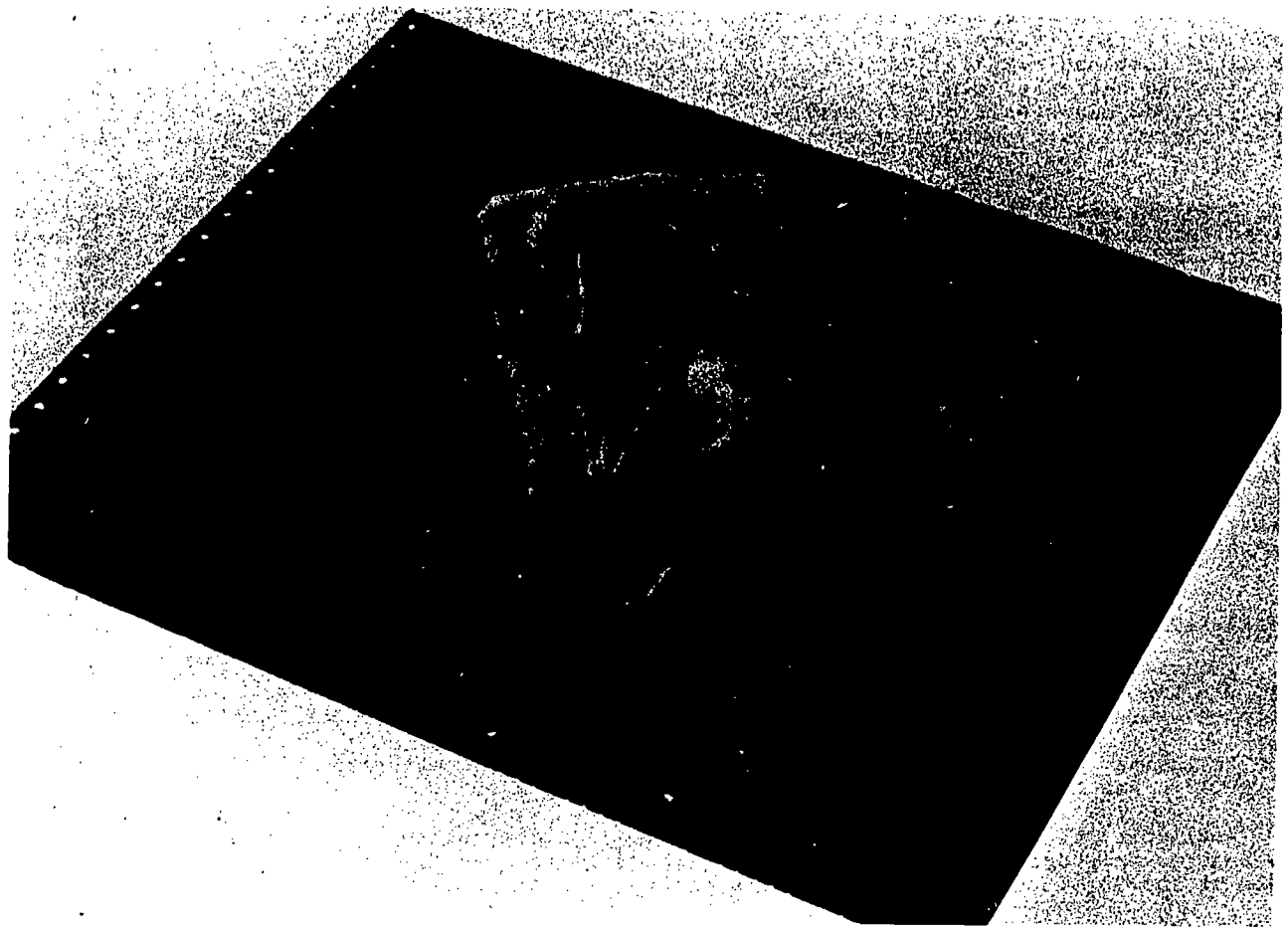


FIGURE 1: Clay model of function $z = x^3 + y^3 - 6xy$.

Students spent many hours empirically examining functions using this technique. Computer generated contour plots were used to imagine the graphs of the functions.

Immediately following this the class was asked to skip directly to the section of the text which had problems on maxima and minima and to do every two variable problem given, empirically, with the use of the computer.

Description of the Computer Programs

In describing a program to be used by students whose familiarity with computers varies widely, three principles should be borne in mind. First, the program must be simple to use. Any information necessary for the operation of the computer program must be contained in a brief, easy-to-read description of the program or, preferably, typed out by the machine as it is needed. Most students who are apprehensive of any device popularly labeled as a "thinking machine" are reticent to absorb any large body of facts before executing a program.

Second, such a computer program should be as foolproof as it can be made. Ideally, no error that a student could make in executing the program should result in the machine either stopping or running on endlessly without indicating what error has been made. Likewise, the student should not be confronted with the computer's waiting for some input without first indicating to the student what kind of input is required.

If, in addition, the computer program is to serve as an efficient learning tool it must be flexible in its operation. That is, it should operate in such a way as to enable a student to "wander" through the relevant subject material in a manner which seems proper to him. If, in the middle of exercising one capability of the program, the student becomes curious about some related material, he should be able to abort the sequential execution of the program and transfer to the part of the program which will allow him to satisfy his curiosity.

The programs described in this paper which work together as a unit were designed with the above three principles in mind. Once the program system called EVALF has been executed, a complete description of the capabilities of the programs and step-by-step instructions as to what to do next are given to the student by way of the computer's console typewriter. All the programs in EVALF are written in FORTRAN and run on an IBM 1130 computer. The student can communicate with the computer both by typing on the keyboard and by using one of the fifteen console switches available on the 1130. Only the function which is being evaluated and/or plotted needs to be entered via the card reader.

The function $Z = P(X,Y)$ is stored in the memory of the computer via the card reader as FUNCTION Z(X,Y). Only one of these functions can be stored for active use.

Upon execution of the plotting routines of EVALF, the student is asked to input values for the upper and lower limits on X, Y, and Z. Then the student is asked to decide whether he wants the large or small plot. The large plot is 30 lines down and 50 characters across with 11 contour lines, while the small plot is 17 lines down and 25 characters across with 6 contour lines). The graph is then produced together with a key for the contour lines. In the small plot the contour lines are drawn using the characters 1 through 6; on the large plot the characters 1 through 9 plus A and B are used (see Figures 2 and 3). It is then possible to replot the same function using different limits or to enter a new function to be plotted.

In most cases the student using EVALF has little or no feeling for the range and domains in which the function is interesting. Our system allows them to search for the regions of interest using the small plot routines and then to use the more time-consuming large plot routines to investigate these regions.

It should be noted that whenever the plotting routines are used, the plotting procedure can be aborted by use of a console switch. This allows the user either to correct mistyped data or, upon seeing that no contour lines are being produced, to switch to another region of space without having to wait for the graph to be completed.

Reactions to the Experience

Before turning to our conclusions, it might be useful to mention some student reactions. The students enjoyed this non-verbal, informal experience. This could be judged not only from their own comments but also from the fact that many of the students spent considerable time outside of class studying functions.

One student commented explicitly on the fact that for the first time she began to see the calculus around her, i.e., to use the ideas to interpret reality, particularly geometric reality.

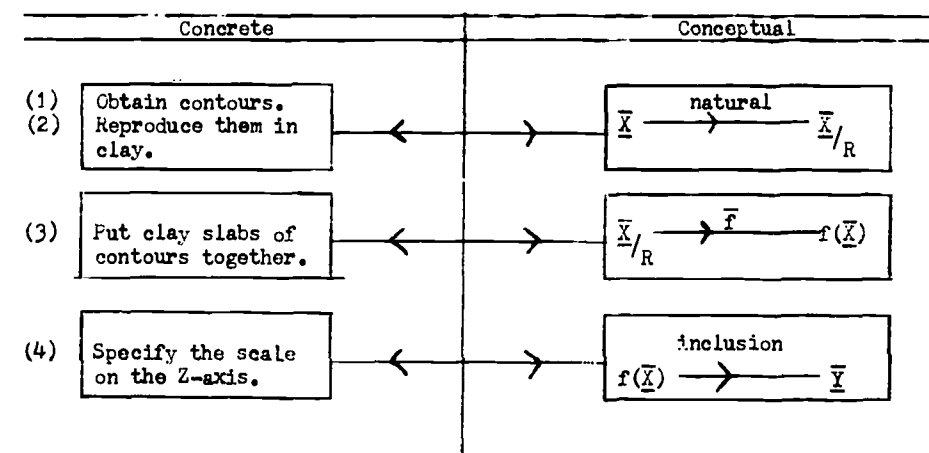
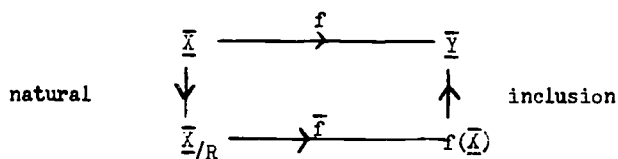
The experience also served to introduce several students (and one of us) to the use of the computer and to stimulate a desire to use it further and understand it better.

Conclusion

The whole experience to which these students were exposed had four stages:

1. Obtaining and studying the contour diagrams of the function.
2. Constructing slab clay models of these contours.
3. Constructing the model of the graph from the models of the contours.
4. Specifying the scale.

The total experience and the four stages described are analogous to one of the basic procedures of contemporary mathematics. In fact, let $f: \bar{X} \rightarrow \bar{Y}$ be a mapping between sets; then the following diagram is commutative:



The fact that the diagram is commutative simply means that f is completely determined by "natural," \bar{f} , and "inclusion."

The aim of the experience was to help students develop their ability to visualize functions depending on more than one variable. So far, 25 students have had this experience and it appears to have been successful. This conclusion is based on student reactions and on an informal comparison of these students with students who have not had this experience.

The students quickly developed the ability to imagine what would happen if stages (2), (3) and (4) were carried out. This meant that only stage (1) needs to be carried out explicitly. The

tedious aspects of this step are quickly and effectively done by the computer. The student is, thereby, able to study many functions, or to study one function extensively. He can be given challenging multivariable non-linear problems and can solve them efficiently. All of this can be achieved prior to the introduction of abstract tools. In fact, this experience paves the way for an effective initiation into the use of more abstract methods.

These computer produced contour plots become one of the most useful and effective tools at the students' disposal. They are used to solve practical problems and to understand new concepts being studied; thereby becoming the bridge between these two activities. Perhaps this is the optimal role for the computer in calculus (or in any other mathematical subject). In this role the computer becomes a versatile and effective way to individualize the learning process.

The wide variety of clay models that a class can produce in a few hours are useful visual aids for lectures or discussions on the calculus of several variables.

These experiences could have wide application. They are accessible to anyone who understands or wants to understand what the word function means. Calculus is not necessary in order to carry out the experiences and solve practical problems. Thus, they are available to students without extensive mathematical background, for example, beginning science students, or high school students. Indeed, the experience would be useful to any student interested in understanding quantities depending on many variables--a wide audience, to say the least.

The usefulness of the models has led us to try to produce some clear plastic models corresponding to useful and interesting functions. However, a number of technical problems in working with plastic remain unresolved. We also investigated the possibility of using holograms as a vehicle for representing the graphs of these functions in three dimensional space. However, the high cost of the master hologram (approximately \$1400) prevented us from implementing this technique.

We are attempting to use similar methods in studying functions from \mathbb{R}^2 to \mathbb{R}^2 (or \mathcal{G} to \mathcal{G}) but as yet have only begun this. Such programs could make beginning linear algebra and multivariable calculus more widely accessible.

APL AT HAMPSHIRE

Everette Hafner
Hampshire College
Amherst, Massachusetts 01002

SUMMARY

Now in its second year of experience with students working in a new curricular framework, Hampshire College has devoted a large effort to drawing its people into computer consciousness, with APL as the language of main interest. The principal facility for the project is UMASS, a time-sharing system based on a CDC-3800 installation at the University of Massachusetts at Amherst. Points of emphasis in our approach are: (1) the use of video tapes and other aids to self-instruction in computer technique, (2) encouragement of independent study through the invention of original programs, (3) applications to the arts, (4) experimentation in mathematics, and (5) simulation of physical phenomena as a complement to laboratory work. This paper presents typical results of the effort so far, with emphasis on examples generated by the author and his students.

We are told of a music professor, new to computer languages, who encountered great difficulty in writing a BASIC program for producing an unending sequence of twelve-tone rows. (A tone row is an ordering of the twelve notes without repetition.) He took his problem to a colleague in physics, who happened to be working with APL at a computer terminal. While he was talking, the physicist casually typed a few lines, tore off the page, and asked the composer to repeat the problem slowly and clearly.

Then he said, "Well, George, this terminal is equipped with a microphone. It tries to understand easy problems and work them out for us. Yours is easy enough. Try typing in the word TONEROW and see what happens." After a little more encouragement, George tried it:

TONEROW

```
6 11 10 7 2 4 3 12 8 9 1 5
9 1 12 11 7 5 8 3 10 6 4 2
6 11 9 2 3 12 7 10 1 5 4 8
5 11 7 3 10 6 8 4 9 12 1 2
12 1 8 2 3 9 7 5 6 10 4 11
11 1 5 3 8 4 9 2 10 7 6 12
4 10 9 3 7 6 8 2 11 5 1 12
8 2 6 7 3 4 5 10 12 1 9 11
```

and so on, until they stopped the program.[1]

Whether it really happened or not, the case illustrates some things about APL that are of first importance to us: its fast access to computation, its mathematical elegance, the good sense of its built in format, its concise structure, and the mnemonic aspects of its character set. This is not to say that the language does not have some clear disadvantages in comparison with, say, FORTRAN. There may be little doubt that for most conventional research problems, with large store of data and highly iterative procedures, APL is not ideal. But it seems to us to be the most deeply rewarding approach to the computer for people whose aim is toward understanding the nature of the computing process itself. One purpose of this paper is to suggest, by example, the several ways in which APL has worked well for us. Another is to invite comment and criticism from groups whose experience may have been different from ours.

A principal challenge of the Hampshire program is the development of useful tools for independent study, especially on the part of students whose preparation and incentive are initially weak. The computer can enter this picture in a variety of ways, many of which (especially the notion of computer-programmed stepwise instruction) have been explored at length. We have a new idea about this. Take a student who is naive in science, yet curious about the ways in which scientists operate. Give him a brief course (we use a set of brief video tapes) on the housekeeping aspects of an APL terminal. But tell him nothing of the language except a few points of convention: how to create variables and arrays. Urge him to imagine an unexplored world of "laws" to which the terminal gives us access. It is his world to study with whatever ingenuity he can muster; it is then, in a tiny sense, a model of the real world.

Most people in this situation work quickly through the easy APL functions and begin to understand how the language is organized. Before long, their experiments develop a scientific flavor. There is indeed a good strategy for investigating an APL function just as there is for

where we have reproduced the first few lines of the long array.

The algorithm used here is conventional, but let us now introduce a new viewpoint. Suppose we ask the student to write an APL function for primes up to N in a way which expresses the mathematical definition of prime most directly. The function should simply choose the set of integers such that each is divisible only by itself and 1. A response to this challenge is:

```

NEATPRIME N
[1] (2=+/[1]0=(1N)0.[1N])/1N∇

NEATPRIME 50

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

```

This function is our favorite example of the power and conciseness of APL. It also illustrates extremely well the close correspondence between the structure of the language and mathematics itself.

The last observation has led us to look at some ways in which a student can lead himself through a branch of mathematics. There are the usual obvious things to do in number theory. But what about modern algebra, for example? We have constructed a heuristic exercise on a nine-element field which lists the addition and multiplication tables and verifies the nine field postulates by testing all possible cases. The whole process is fast and simple in APL. Here, for example, are the tests of the two associative laws:

```

^/ ^/ ((X A X) A X) = X A (X A X)
1
^/ ^/ ((X M X) M X) = X M (X M X)
1

```

where A and M are functions for addition and multiplication over all elements of the field X . The exercise continues with a proof of the quadratic formula, after discovering that there are only 45 (out of a possible 72) solvable quadratic forms in the field. We prove the formula easily by showing that it generates all possible pairs of roots. The student is left with a final thought: "It might be interesting to suggest that theorems which are proved on a finite field can serve as conjectures to be investigated on other fields, even including the field of real numbers."

The problem in algebra, carried out entirely by computation, is entirely free of formal proof. It tends for this reason to raise eyebrows, and even to sound alarms, among mathematicians (including students in love with the abstract). But we find the idea powerful and vivid: it can give an illuminating first view of a mathematical structure, perhaps as preliminary to a more formal study. Another possibility is that, through conjecture and test, it can produce new knowledge.

Our best example of actual mathematical discovery came about in an exercise on reciprocals of primes. It is well known and easy to prove that if p is a prime whose reciprocal has period $p-1$, then the first $p-1$ multiples of the corresponding integer permute cyclically. For instance:

```

7
0.1428571428
6 1p142857x16
142857
285714
428571
571428
714285
857142

```

The set of primes with this property is 7 17 19 23 29 47 59 ... and is usually found by search for reciprocals of appropriate period. We wrote a function for arbitrarily large reciprocals:

```

∇ Z+P REC B;Q;R;S
[1] C+1~1+R+1
[2] C+Q,S+LR×B≠P
[3] R+(R×B)-S×P
[4] +2×1(P-1)>pZ+Q
∇

```

and found, for instance, that the reciprocal of 61 produces the 60-digit number

016393442622950819672131147540983606557377049180327868852459

whose first 60 multiples are necessarily cyclic. The APL function forced upon us an awareness of base 10 in computation, and led us to a crucial question. Does the cyclic property of a prime depend on the number base? Suddenly one realizes that the reciprocal of 2 expressed in any odd base is a trivial case, and a few trials with other primes quickly raise the conjecture that for every prime there is an appropriate base. A few more steps, and a little number theory, produce the central theorem for this problem:

The reciprocal of any prime p expressed in base b is cyclic if and only if b is a primitive root of p .

(A primitive root of p is a number whose $(p-1)$ st power is the smallest congruent to 1 modulo p .) The problem has been solved.

We come next to a conjecture in statistics. Consider

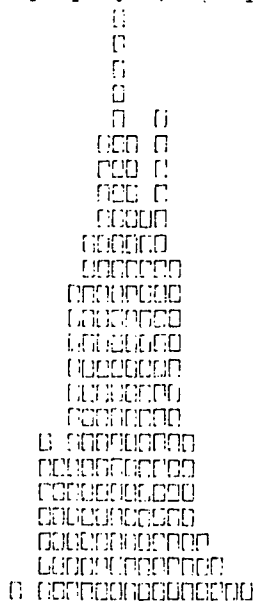
```

7 2+1 GAUSS 166
[1] 2+2*(21,0)0(,(2+4)0.>10+1/A++/(121)0.=L0.5+(+/2(,"")021)0)'\171

```

4 GAUSS 166

0 0 1 0 7 6 13 15 19 24 19 16 20 14 7 3 2 1 1 0 0



This beautifully concise function computes M rounded means of M random integers from 1 to 21 and plots their histogram. (The example is typical for samples of means of 4 integers in the domain.) One observes that the distribution has central tendency at 10, and grows sharper with increasing M . We suggest to the student that the frequencies are distributed normally. He studies the trend of variance with M and attempts to justify it in a formal way.

We like to raise questions suggested by the history of computation. In 1914, Ramanujan discovered that the rational number $2143/22$ approximates the fourth power of π to about one part in three billion. It is natural to ask how lonely this number is in the class of fractions running through the integers to 10,000 and approximating any of the first four powers of π .

Hence:

```

▽ N SEARCH N;I;J;K;L;R;T
[1] PI+01
[2] I+1
[3] J+1
[4] L+K+1[ [J≠PI*I
[5] +((P>3.141592)^(R+(J≠K)*I)<3.141593))/SHOW
[6] +((N+K+1)=I+1)/5
[7] +(N>J+J+1)/4
[8] +(N>I+I+1)/3
[9] →0
[10] SHOW:→((J GCD K)≠1)/6
[11] T+(J≠K)*I
[12] J,K,I,T,(10*-2)×[0.5+(10*10)*T-PI
[13] →6

```

```

▽ Z+N GCD N
[1] Z+1
[2] N+M|N
[3] H+Z
[4] →0*!

```

4 SEARCH 10000

355	113	1	3.141592921	26.69
9891	319	3	3.141592418	⁻ 23.52
2143	22	4	3.141592653	⁻ 0.09

where the first two columns reveal the integer pairs, the third gives the power of pi approximated by the fraction, and the fifth is the error in the eighth decimal place. We see that there are only three cases in which the error is less than about 50, and that Ramanujan's number is best by a large margin. Did he actually perform a search like this (without a fast computer) or was he simply lucky?

We give two examples of our approach to computer-assisted instruction. It is our view that the most efficient mode is simply a brief test of one's ability to carry out a task. Here is a program on orders of magnitude:

```

▽ PRODDER N;P;C;Z;T;A
[1] 'ESTIMATE THE FOLLOWING PRODUCTS, TO THE NEAREST HIGHER'
[2] 'OR LOWER POWER OF 10. TYPE THE APPROPRIATE EXPONENT.'
[3] 'FOR EXAMPLE, IF THE VALUE IS ABOUT 250, EITHER 2 OR 3'
[4] 'IS ACCEPTED. YOU HAVE 15 SECONDS.'
[5] C+1+P+0
[6] DELAY 20
[7] PR2:Z+(250100)×10*-4+P5p5
[8] 'PROBLEM ':C
[9] Z[1];'*';Z[2];'*(';Z[3];'*0.5)*(';Z[4];'*2) ÷ ';Z[5]
[10] Z+100Z[1]×Z[2]×(Z[3]+0.5)×(Z[4]+2)÷Z[5]
[11] T+I20
[12] A+T
[13] +SI×1(I20)>T+2500
[14] +((A<|Z),(A>[Z]))/SM,LG
[15] 'YES.'
[16] B+B+1
[17] +MPLP
[18] SM:'TOO SMALL. VALUE IS ABOUT 10*';([10×Z)÷10
[19] +MPLP
[20] LG:'TOO LARGE. VALUE IS ABOUT 10*';([10×Z)÷10
[21] +MPLP
[22] SL:'TOO SLOW.'
[23] MPLP:→PR2×(N>C+C+1
[24] 'YOUR SCORE IS ';B;' OUT OF ';N;' ';[100×P+N;' PERCENT.'

```

PROBLEM 5
 ESTIMATE THE FOLLOWING PRODUCTS, TO THE NEAREST POWER
 OF LOWER POWER OF 10. TYPE THE APPROPRIATE EXPONENT.
 FOR EXAMPLE, IF THE VALUE IS ABOUT 250, EITHER 2 OR 3
 IS ACCEPTED. YOU HAVE 15 SECONDS.

PROBLEM 1
 $4.2 \times 540 \times (4.7 \times .5) \times (290 \times 2) \div 0.18$

9
 YES.

PROBLEM 2
 $0.05 \times 9.6 \times (750 \times .5) \times (0.056 \times 2) \div 9$

3
 TOO LARGE. VALUE IS ABOUT $10 \times 10^{-2.3}$

and so on.

The second test criticizes one's attempts to rearrange and solve a linear equation, which the program sets up at random. The APL function yields 1 for every true relation in X (including identities), and 0 for everything false:

TUTOR
 WANT INSTRUCTIONS? TYPE Y OR N.
 N
 $((2 \times X) + (7 + 3 \times X)) = (-1 + X + 3 \times X + 3)$
 $((4 \times X) + 7) = (X - 1)$
 1
 $X = +8 \div 3$
 0
 $X = -8 \div 3$
 1
 X
 -2.666666667

TUTOR
 WANT INSTRUCTIONS? TYPE Y OR N.
 N
 $(X + X + 2) = (X + X)$
 $X = 0$
 1

TUTOR
 WANT INSTRUCTIONS? TYPE Y OR N.
 N
 $(1 + (-1 \times X) + (-1 + 2 \times X) \div 5) = (-3 + (8 \times X) + (5 + 9 \times X) \div 5)$
 $((-3 \times X \div 5) + 6 \div 5) = ((31 \times X \div 5) + 4)$
 1
 $X = 17 \div 13$
 0
 $X = 13 \div 17$
 1

(The program for this exercise is available on request.)

We have built a library of interesting mathematical sequences, mainly for the purpose of studying their rates of convergence. Here is a familiar sequence converging to the square root of any positive number:

```

W 1 500 0
[1] +((I*(I+Y+0.5*Y+X*Y)-X*0.5)>10*-15)/1

```

1.4 000 2

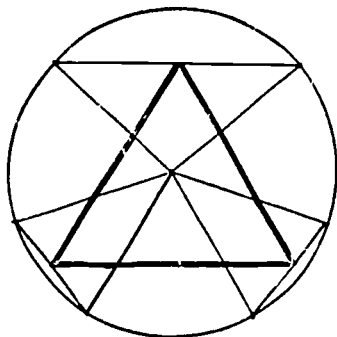
```

1.41421356237468714
1.41421356237468714
1.41421356237468714

```

The accuracy doubles at each step.

We have encountered some excellent examples of conjecture and experiment in geometry.



One of the best has led us to the theorem illustrated here: the midpoints of the three chords joining arbitrarily chosen 60-degree arcs of a circle are vertices of an equilateral triangle. Before attempting proof, which is straightforward but arduous, we composed an APL function which returns, to 14 significant figures, the lengths of the three segments determined by a pair of angles, in radians, arbitrarily chosen for each test.

Typical results:

TRI 1 2	1.3593192074603	1.3593192074603	1.3593192074603
TRI 1 3	0.94802829331059	0.94802829331059	0.94802829331059
TRI 2 3	0.60594333139318	0.60594333139318	0.60594333139318
TRI 3 3	0.12721726366189	0.12721726366189	0.12721726366189

quickly bolstered our confidence in the truth of the conjecture.

We have been using APL with good success in simulation of physical phenomena. The following function, composed for a course in acoustics, yields the progress of longitudinal waves on a lattice of identical masses and springs:

```

? WAVE C[1]; T; Y; V; A;
[1] V+(1+V+C[1])DT+0
[2] V+Y+
[3] GO:V+(1φY)+(-1φY)-2×Y
[4] Z[1]+Z[1+1]+0
[5] Y+Y+DT×V+V+Z×(DT+C[2])×M×(M-1)+25
[6] +(~V/(T+T+DT)=A+0.5×1G[2]÷0.5)/AUFAD
[7] M=M,Y
[8] +GO
[9] AUFAD:=(C[2]>T)/GO
[10] V+φ(((ρV)÷"+1),"+1)ρV
[11] M+0,A,,10.5+100×M
[12] φ((M+2),((ρV)÷M+2))ρM
V

```

WAVE 15 10 .1

11:

	0	.5	1	.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	.31	.15	.41	.41	.9	1	0	0	0	0	0	0	0	0	0	0	0	0
1.0	0	-.47	-.27	-.4	-.34	.47	.19	.4	0	0	0	0	0	0	0	0	0	0	0
1.5	0	-.24	-.43	-.32	-.18	.20	.47	.29	.9	2	0	0	0	0	0	0	0	0	0
2.0	0	.16	-.12	-.40	-.36	-.29	.4	.42	.38	.16	.4	1	0	0	0	0	0	0	0
2.5	0	-.8	.15	.2	-.33	-.37	-.36	-.12	.32	.42	.24	.8	.2	0	0	0	0	0	0
3.0	0	.4	-.10	.10	.12	-.23	-.35	-.39	-.27	.17	.4	.31	.14	.4	1	0	0	0	0
3.5	0	-.2	.6	-.10	.3	.18	-.12	-.30	-.39	-.39	0	.36	.37	.20	.7	0	0	0	0
4.0	0	0	-.3	.7	-.8	-.4	.19	-.2	-.23	-.35	-.45	-.17	.25	.38	.23	0	0	0	0
4.5	0	0	1	-.4	.8	-.4	-.10	.16	.7	-.15	-.29	-.47	-.34	.4	.18	0	0	0	0
5.0	0	-.1	0	.2	-.6	.7	1	-.13	.11	.13	-.8	-.25	-.54	-.67	-.42	0	0	0	0
5.5	0	1	-.1	-.1	.4	-.7	.5	.5	-.15	.2	.9	-.15	-.42	-.71	-.66	0	0	0	0
6.0	0	-.1	1	0	-.2	.5	0	.6	-.23	-.25	-.19	-.24	-.8	.10	0	0	0	0	0
6.5	0	1	-.1	1	0	-.4	.4	-.11	-.18	-.15	-.42	-.31	.24	.56	.45	0	0	0	0
7.0	0	-.1	1	-.2	-.1	-.2	-.1	-.14	-.32	-.38	.2	.26	.38	.41	.17	0	0	0	0
7.5	0	0	-.2	-.1	-.7	-.14	-.21	-.37	-.20	.6	.31	.58	.27	-.15	-.9	0	0	0	0
8.0	0	-.2	-.3	-.10	-.16	-.28	-.32	-.10	.15	.51	.47	1	-.3	1	-.18	0	0	0	0
8.5	0	-.4	-.13	-.21	-.31	-.25	1	.31	.56	.27	-.1	-.4	-.18	-.3	.19	0	0	0	0
9.0	0	-.13	-.24	-.31	-.17	.11	.44	.49	-.15	-.4	-.15	-.10	.10	.3	-.3	0	0	0	0
9.5	0	-.17	-.22	-.4	.26	.51	.38	-.6	-.11	-.15	.2	.4	.2	.5	-.12	0	0	0	0
10.0	0	.3	.21	.45	.54	.27	-.4	-.16	-.8	.4	1	.9	-.5	-.14	.18	0	0	0	0

In the example, a deformation at the left end of a 15-element system returns with opposite sign (and some distortion) after reflection from a fixed end. On the corresponding continuous system, reflection would occur in 10 units of time.

Looking to the uses of computers in the arts, we have worked with APL as a tool for composition of music, as well as for study of problems in musical acoustics. It has been interesting, for example, to compose a set of functions for examining divisions of the octave in equal temperament. We then carried out a search for tempered scales which contain acceptable diatonic intervals. If one gives high weight to perfect fourths and fifths, the first few good scales have divisions of 12, 17, 19 and 22. If we include the two musical thirds as well, the sequence begins with 19 and 31. The 19-tone scale is especially attractive for the perfection of its minor third, which it produces with an accuracy of 0.01 per cent. We have therefore learned to set the keyboard of an electronic synthesizer for performance of music written for this scale.

The principal conclusion from our experience so far is that APL lends itself with good effect to an extraordinary variety of tasks and challenges. It has therefore been a major contributor to the development of programs in a new experimental college.

There is, however, a large unanswered question which we set before ourselves in early stages of science planning at Hampshire. To what extent can wise use of the computer make it possible for a student of science to avoid, or at least to delay, a major investment in the formal study of applied mathematics? We expressed an early hope for this in the following way :



"It was once necessary for students of any quantitative science to acquire fundamental knowledge of the group of mathematical disciplines (mainly in analysis, ordinary and partial differential equations, probability, and theory of functions) whose applications form the body of theoretical science. But we are now in a position to persuade our students that modern computer science exposes much of applied mathematics as a set of programmable procedures whose theoretical foundations can be left to specialists. Indeed, it has become increasingly clear that a computer-based approach to applied mathematics is more versatile in application than the approach through pure mathematics. Students of limited mathematical talent need no longer confine themselves to the simplest, analytically most manageable (and often least interesting) examples of a scientific problem. The computer is their servant in displaying wide variety of example within a simple mathematical formalism."^[2]

The statement is perhaps too optimistic. We think not. But more important is the question of the philosophy itself. Some scientists (especially mathematicians) react with horror to the idea. Others see it as at least worthy of exploration. We take the view that the pedagogical feasibility of fast and elegant computation is far from known, and that institutions like Hampshire must devote part of their effort to the question.

The author takes pleasure in acknowledging the assistance of many colleagues, including especially Kenneth Hoffman, Kenneth Iverson, Howard Peele, Michael Ubell, Rebecca Wills, Conrad Wogrin and John Wright. Computer facilities at the University of Massachusetts, SUNY at Binghamton, and the IBM laboratory at Poughkeepsie were essential to our project. Financial support for part of our work was provided under a grant from the Esso Education Foundation.

REFERENCES

1. What the physicist had typed in was simply

VTNEROW
[1] 12712
[2] --- 1v

2. E.M. Hafner, Arches of Knowledge, Hampshire College Planning Bulletin 10, August 1969.

COMPUTER GENERATED PICTURES FOR TEACHING CALCULUS

Roger B. Kirchner
Carleton College
Northfield, Minnesota 55057
Telephone: (507) 645-4431

In a paper given at the Iowa Conference [1], the author presented programs for mathematics instruction which could be run from an ordinary time-sharing terminal. The programs were classified as investigative, instructive, and illustrative. This classification continues to be useful. In this paper, we emphasize the use of 35 mm slides that can be taken of the graphical output of programs of these three types.

The computer system used consists of a PDP-8/L computer with 8K of memory, an ASR-3J teletype for input-output, and a KV8/L graphic display system with a Tektronix 611 Storage Oscilloscope. Since the storage scope is suitable for viewing by only a small number of students at a time, it was decided that the best way to communicate the results of programs with graphical output was by means of 35 mm slides. Initially, the main purpose of the slides was to motivate students to use the instructional programs. However, in showing the slides to both ordinary and computer-oriented sections of calculus, they were found to be useful by themselves. Sequences of slides can simulate the experience of viewing the generation of pictures on a storage scope, and they can improve the direct experience in cases where more than a few seconds is needed to generate each picture.

It does not seem to be critical that the student sees the pictures being generated on a scope. It is sufficient for him to know that he could duplicate the pictures if he had access to a terminal with a scope. Although the student loses the freedom to vary parameters when he can only see the slides, this is less of a disadvantage if there is a rich variety of slides available. It is thus hoped that, by duplication, the slides produced can be of use to instructors in schools where graphics capabilities are not available.

Investigative programs allow a student to study a class of procedures such as root finding methods, or a class of mathematical objects such as rational functions or implicitly defined curves. A specific example is TAYLOR, in which the user has the option of studying Taylor polynomials for any of nine different functions. With some, he can specify the point about which the expansion is made. He can specify the domain, the range (scale equal to the domain scale, specified maximum, or computed maximum), and the polynomials he wants to plot. The computer then sketches the function and the approximating polynomials. Photographs T-1 through T-5 show some approximations to $\sin x$ near $x = .2$. One observes that the $2k-1$ st and $2k$ th approximations are very much alike. But the approximations are polynomials. Consider T-4. Can an 8th degree polynomial look like a seventh degree polynomial? The user will want to sketch another picture of these polynomials using a larger range and domain.

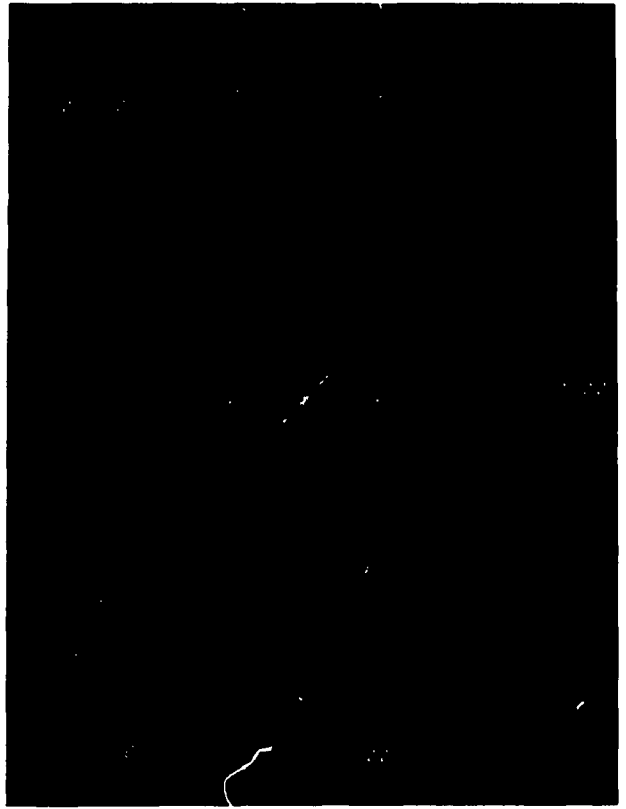
Instructive programs are designed to introduce and reinforce a mathematical idea such as the limit concept, the definite integral, or the fundamental theorem of calculus. Pictures have been included from three instructive programs, YOUEP ("You Epsilon Me Delta"), MEEP ("Me Epsilon You Delta"), and PNDTHM.

YOUEP and MEEP are games originally programmed to run at a teletype, and were described in the Iowa paper. Playing them on the scope is much faster, and they are much more instructive when a graph of the function being studied is sketched. First the rules are stated. Then, $f(x)$, x_0 , and L are chosen by the user. The computer makes a judgment about its chances of winning. In YOUEP, if the computer thinks it can win, it will use an estimate for a Lipschitz constant at x_0 to produce a delta for any inputted value of epsilon. If it thinks it will lose (the estimate for a Lipschitz constant is large), it will resignedly try three successively smaller values of delta for each inputted epsilon, and thus force the user to play carefully. Photographs Y-1 through Y-14 show a sample game of YOUEP. $f(x) = x \sin(1/x)$, $x_0 = 0$, and $L = 0$. The limit is L , and the computer senses this. When a value of ϵ is inputted, the computer sketches the graph of f near x_0 , draws a horizontal strip of width 2ϵ centered at $y = L$, chooses a value for δ (in this case $\delta = .900719\epsilon$ which is clearly small enough), draws a vertical strip of width 2δ centered at x_0 , and waits for the user to input a value of x . The result of the users response is graphed, and the relevant calculations are made. In this example, the computer eventually wins.

Part of a short game of MEEP is shown in photographs M-1 through M-3. Here the roles of the players is reversed. Note that the computer has found a winning choice for ϵ . The games are actually a little more fun to play when the user can win. Then the user can play a little sloppily and test whether the computer can take advantage of his mistakes. A significant feature of these games is that the user does not have to think about scaling. That is done automatically, and he can concentrate on the ideas. It may be useful to modify the programs so that the point of the games is to find winning formulas for ϵ , δ , or x . They would then have an investigative rather than an instructive flavor.



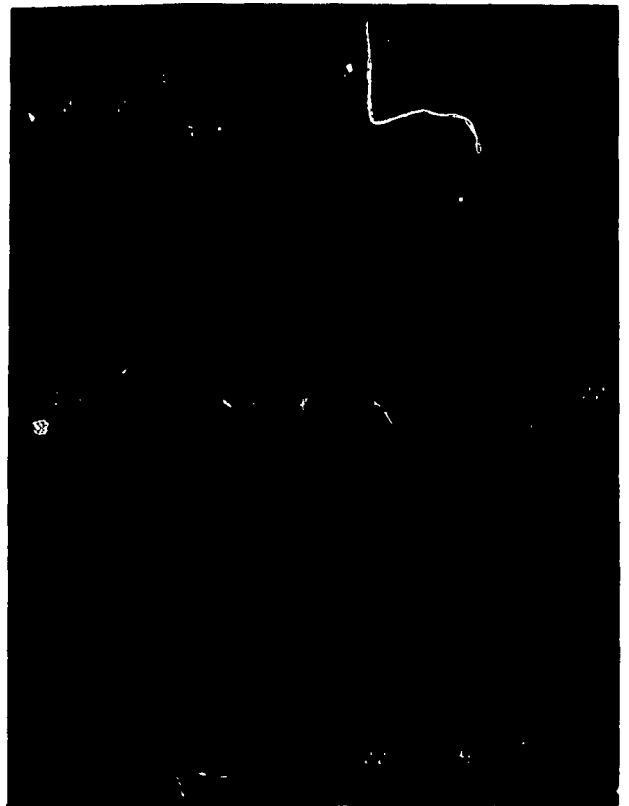
T-1



T-2

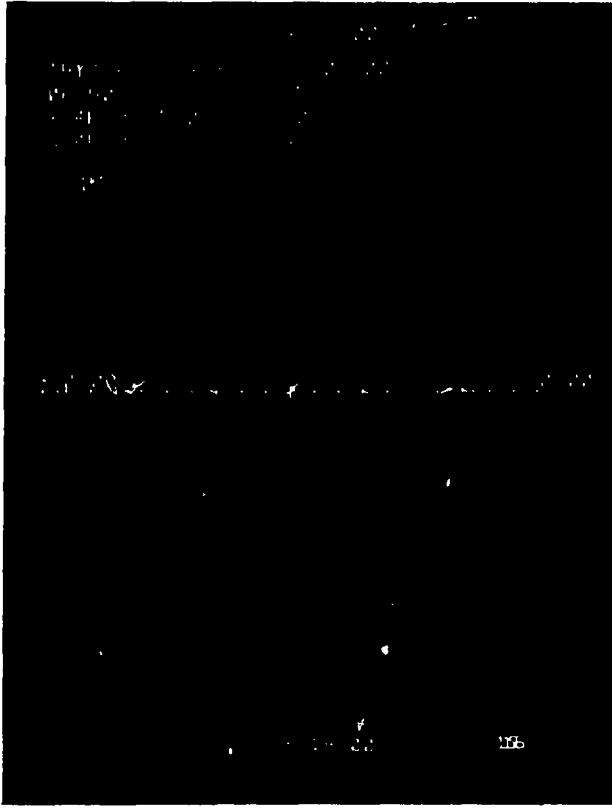


T-3



T-4

360



T-5



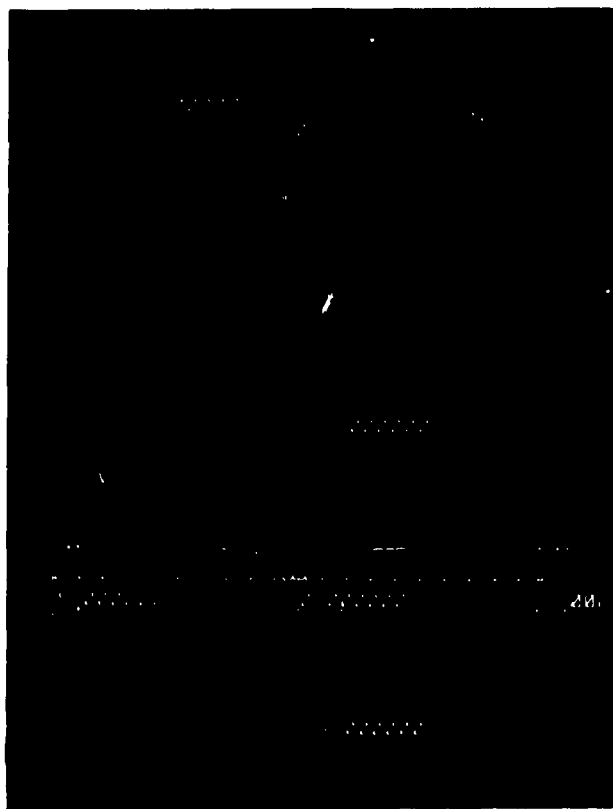
Y-1



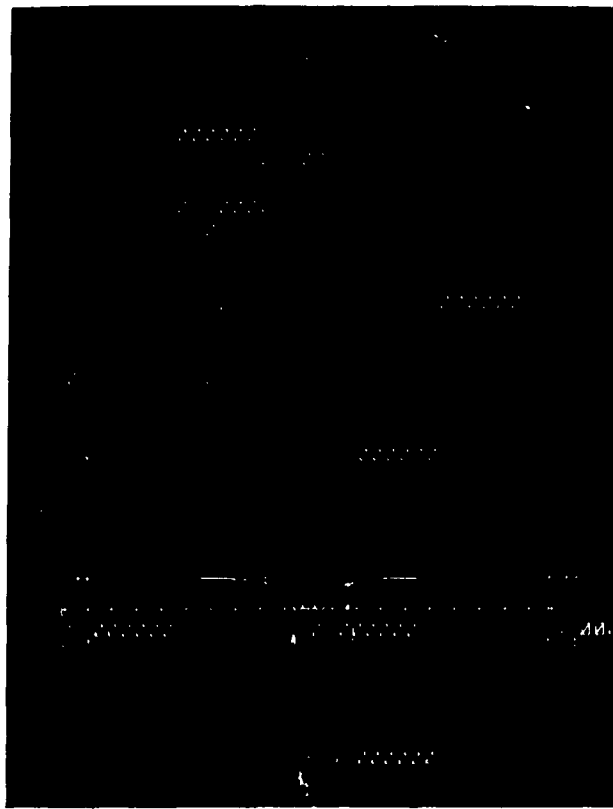
Y-2



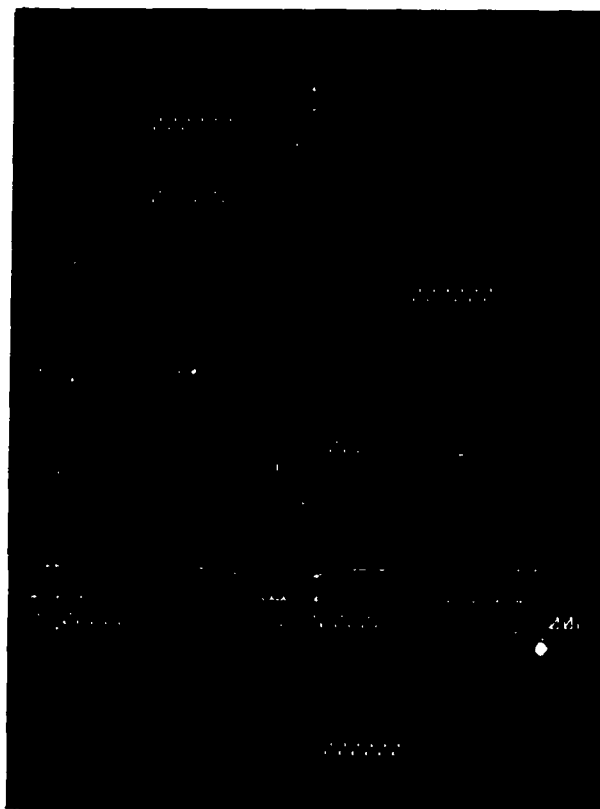
Y-3



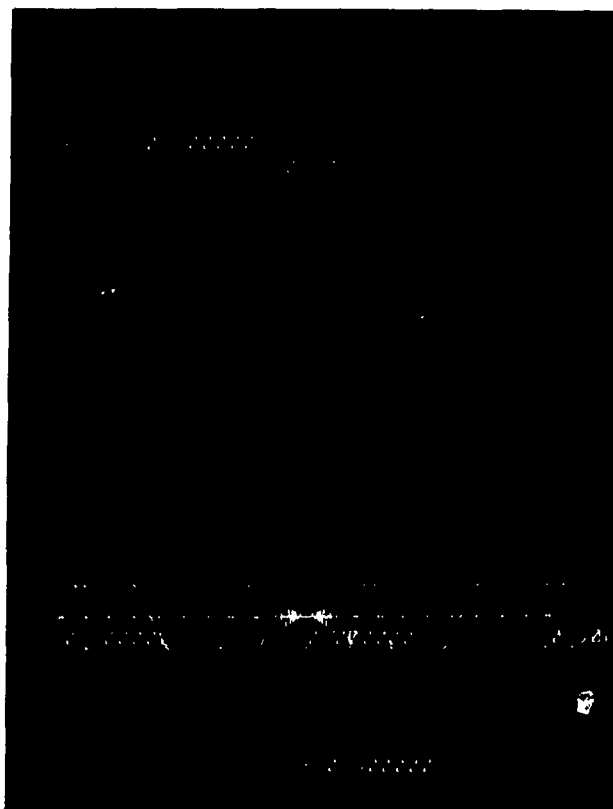
Y-4



Y-5



Y-6



Y-7



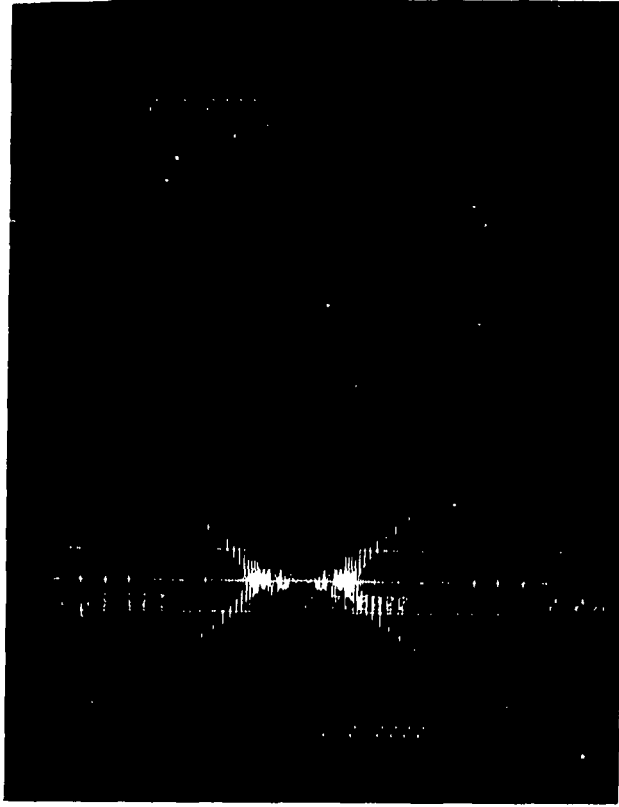
Y-8



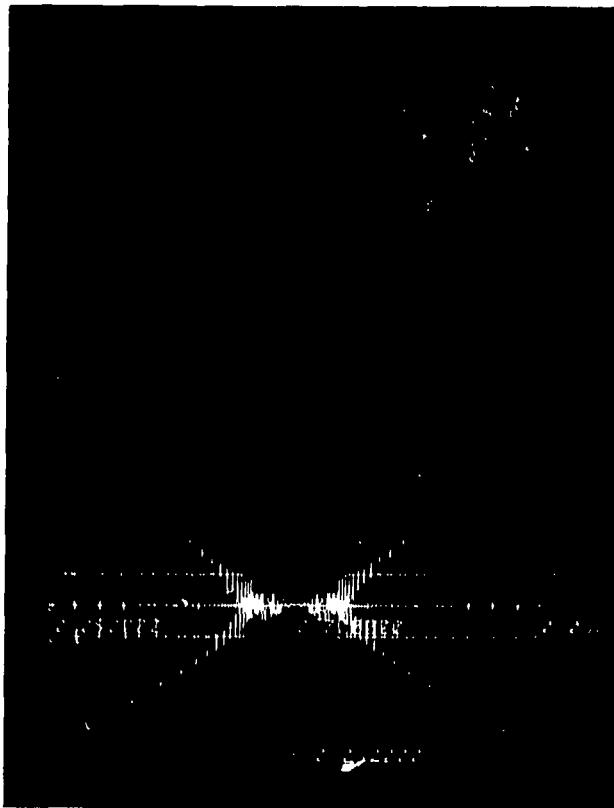
Y-9



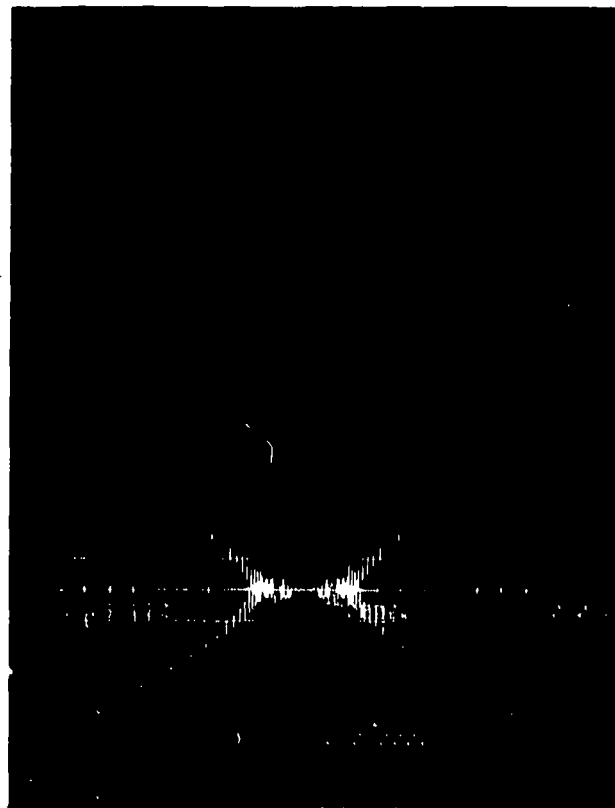
Y-10



Y-11



Y-12



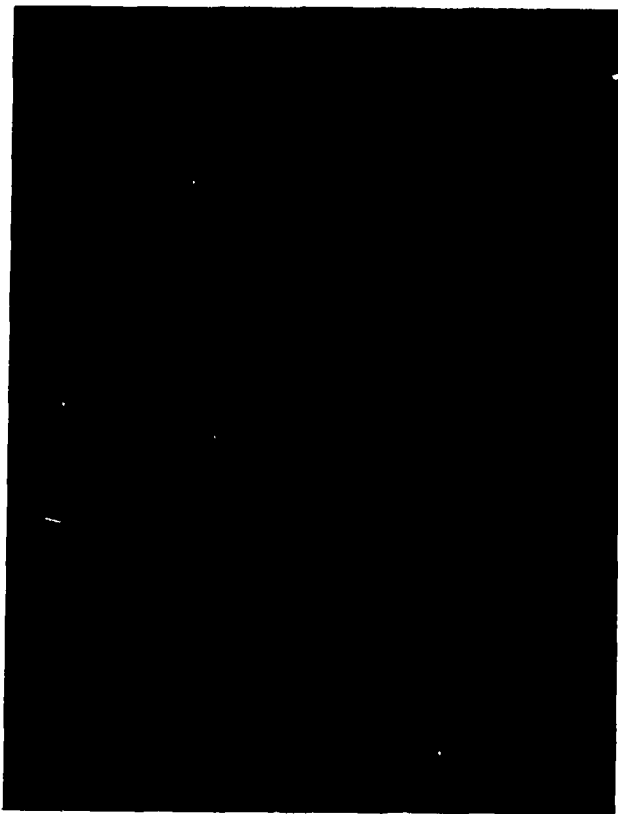
Y-13



Y-14

364

364



M-1



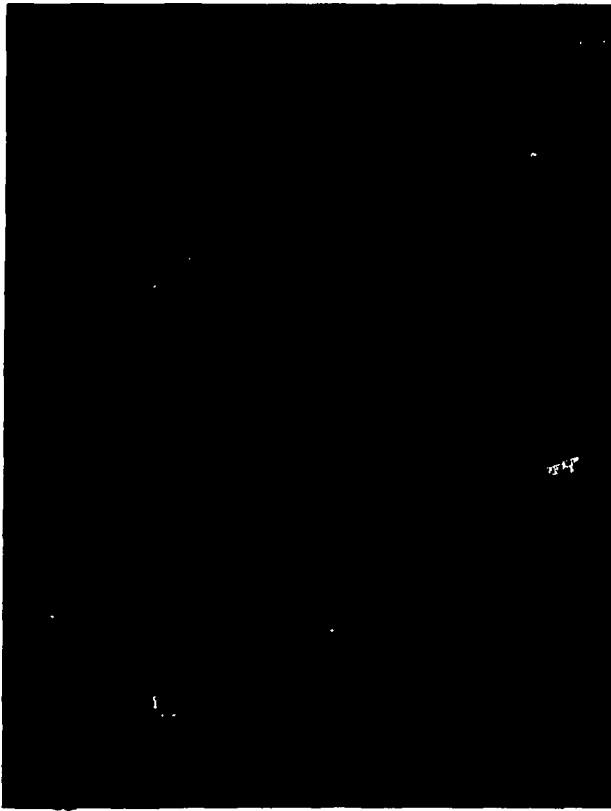
M-2



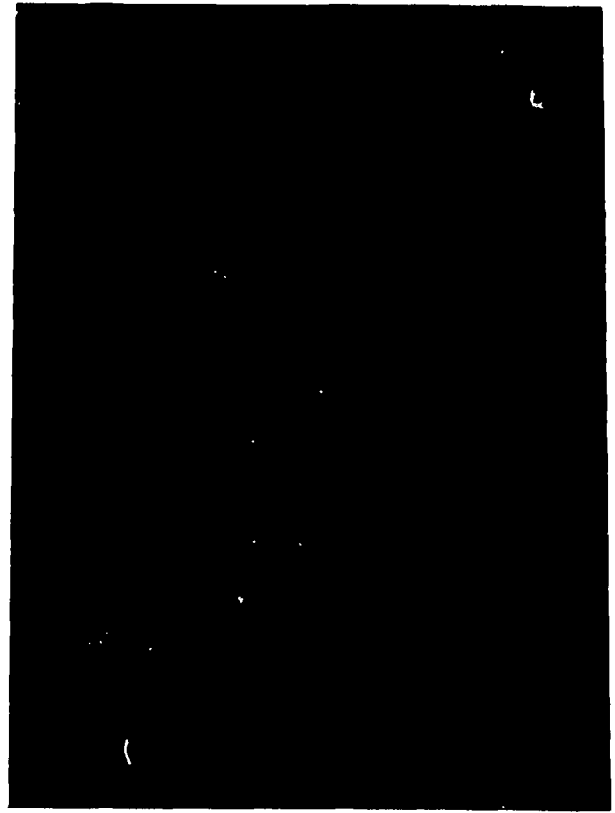
M-3

365

365



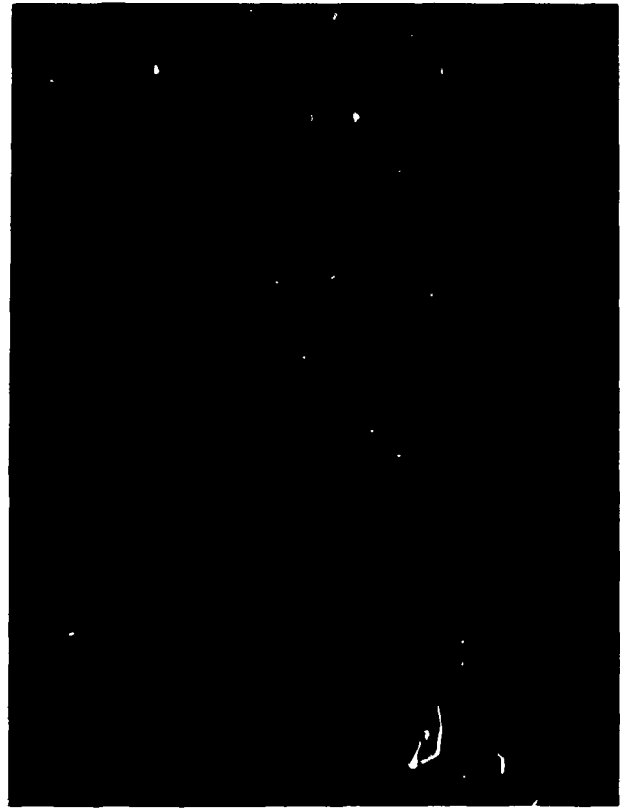
F-1



F-2



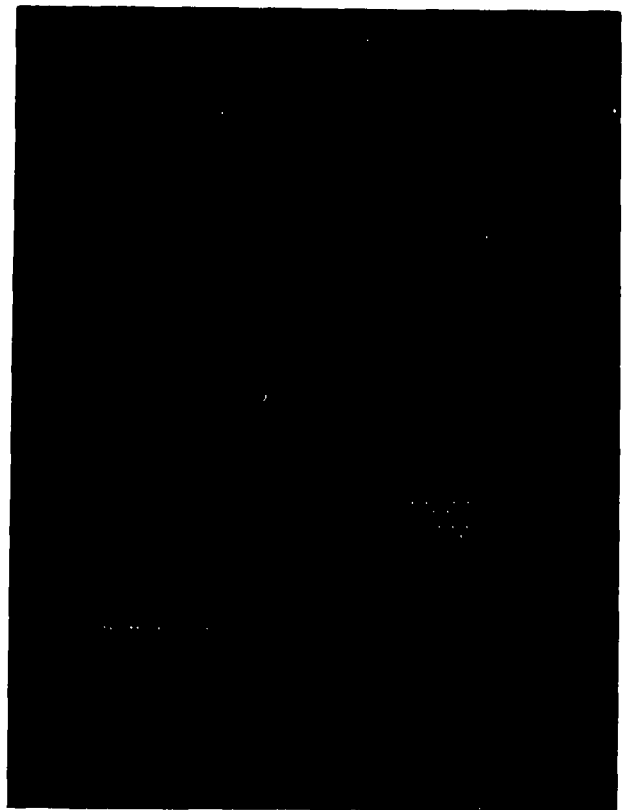
F-3



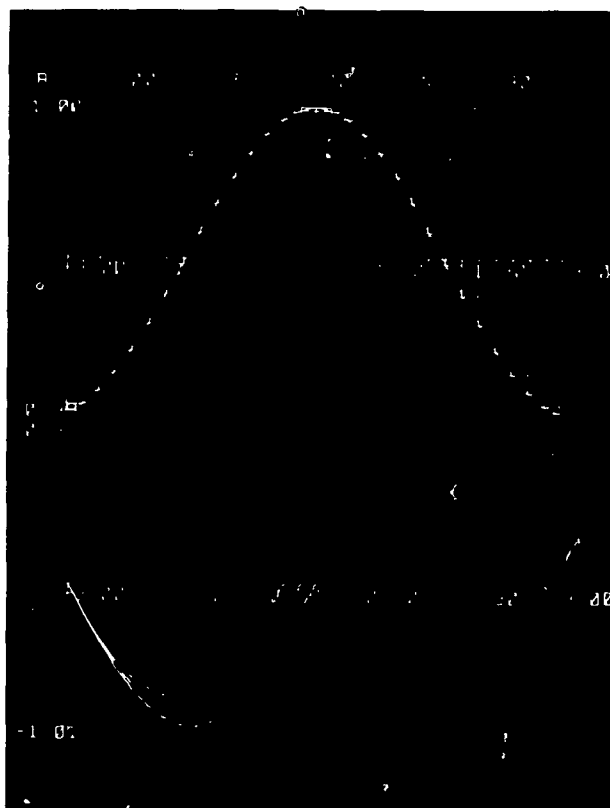
F-4



F-5



F-6



F-7

PNDTHM is a program designed to verify the fundamental theorem of calculus. The user chooses one of ten functions, an interval $[a,b]$, and a positive integer n . On the upper half of the scope, the computer sketches the graph of f on $[a,b]$. On the lower half, it sketches the graph of $F(x) - F(a)$, where F is an antiderivative of f . Then, for i from 1 to n , it draws rectangles which define lower and upper sums for f on $[a,x_i]$ and plots values of these sums on the lower graph, where $x_i = a + (b-a)i/n$. The graphs of the lower and upper sums are made piecewise linear. Pictures P-1 and P-3 which attempt to explain the situation are unsuccessful because the description is not quite accurate and there is too much information on each picture. The programmer was too involved in getting the computer to draw sigmas and integrals. However, one can see from the definition of $P_{MIN}(I)$ and $P_{MAX}(I)$ in P-3, that the sums are lower and upper sums only when the local extrema of f are subdivision points. Some simplification like this was unavoidable.

The development of the graphs is quite interesting to watch. It is almost movie-like. One observes that $F(x_i) - F(a)$ is always between the lower and upper sums for f on $[a,x_i]$, and is led to the conclusion that $\int_a^x f(t)dt = F(x) - F(a)$. See pictures P-5 and P-7.

Illustrative programs demonstrate unusual examples or counter-examples. Examples are the graphs of approximations to the Cantor function, to a non-differentiable function, and to a space filling curve. Unfortunately, because of limited access to enlarging and printing facilities, no prints of these pictures are available for this paper.

The photographs were taken with a SLR camera using Kodak High Contrast Copy film (ASA 64) and an exposure of 5 seconds at $f5.6$. The slides are thus black and white and project black on a white background.

REFERENCES

1. "Programs for Computer Extended Instruction in Mathematics," Proceedings of a Conference on the Computer in the Undergraduate Curricula (Iowa City: University of Iowa, 1970), pp. 4.37-4.44.

COMPUTER-BASED EDUCATION LESSONS FOR UNDERGRADUATE QUANTUM MECHANICS

Carol D. Bennett
University of Illinois
Urbana, Illinois 61801
Telephone: (217) 333-6212

Simulations using a computer to plot quantum-mechanical wave functions for a chosen potential have been popular in physics education. They give the student the opportunity to experiment with the behavior of wave functions without the distraction of complicated calculations. Such simulations have been expanded and lessons written on the PLATO system at the University of Illinois. Brief descriptions of new lessons are given as well as how these and others have been integrated into undergraduate physics courses. A new lesson on the two-electron helium atom and a self-consistent calculation of the lowest-energy electron wave function is described in more detail.

Science and engineering students work out the infinite square-well problem in detail in the third-semester elementary physics course. A computer simulation is then used as part of a lecture demonstration to show the behavior of quantum-mechanical wave functions for a finite well. As the well is made deeper, they are able to identify the solutions for the infinite well. As the well becomes shallower, more of the wave function is seen to creep into the classically-forbidden regions.

As part of this lecture demonstration, using an on-line computer graphics facility, the students suggest energy values in the search for the bound state solutions. Before they are able to solve finite-well problems analytically, they get a feeling for the boundary conditions that must be satisfied. This is usually a one-shot event at the elementary physics level and the students' response is highly enthusiastic. Interest is stimulated in how the equations are integrated numerically and how the bound state energies could be found analytically.

The next quantum mechanics course, for juniors and seniors, uses the computer lessons as an integrated part of the course. Towards the beginning, the students individually work through a lesson on phase and group velocity. The main features of this lesson are two "labs", one on phase velocity and one on group velocity, in which the students can vary wave numbers and angular frequencies to see a single wave or sum of two waves plotted at consecutive time intervals. Before entering the first "lab" on phase velocity, the student is guided through questions and exercises on the concepts of wave number and wavelength. At one point a sine wave with a randomly chosen wavelength is plotted along a marked scale. The student is asked for the wavelength and wave number, and must get three correct in a row to proceed. After each "lab" a quiz tests the student's understanding of the relationships between quantities involved in the phase or group velocity of a wave and the general shape of the wave. Key words in the questions such as "increasing" or "decreasing" (e.g., "Increasing k has what effect on phase velocity?") are chosen randomly because the student may have to repeat the quiz after additional experimenting in one of the "labs."

Two later sessions with the computer are directly connected with homework assignments. Some of the problems are listed in the Appendix and could be used with any computer lesson that allows a variety of potentials to be specified by the student. Such guided exercises greatly add to the educational value of these lessons. One set of problems deals with potential wells and bound-state solutions, and the other set concentrates on barrier transmission problems. A favorite problem is the analogy with the optical case of a nonreflective coating. Until they see the wave function plotted for the conditions they calculate, some students do not believe that there can be 100% transmission when the incident wave must pass over a change in potential. Students typically work in pairs at the terminal on these homework problems. The interaction between students during these sessions promotes added awareness of what is happening and increases the amount of investigation into finer details of the problems. They ask each other questions and then work together with the computer to find the answers.

Not all of the potentials studied using the computer are symmetric. For this reason, the lesson that plots wave functions for square and asymmetric wells starts integrating from the far left with zero amplitude and a slight slope, and integrates to the right. When the amplitude at the far right becomes very small, a bound state can be identified. The barrier transmission lesson also integrates starting at the left and plots the incident plane wave, the reflected wave, and the transmitted wave. The sum of these is then plotted to show that the total wave function matches properly at the boundaries. For the elementary students, however, a more general lesson is used that leaves the starting point of the integration up to the user. Building the symmetry of the square well into the wave function by starting at the center of the well with a positive amplitude and zero slope for symmetric solutions and zero amplitude and positive slope for antisymmetric solutions is less confusing for the elementary students.

An additional lesson was written for students to use before they start the lessons that plot wave functions. It introduces them to the types of functions that are plotted in various

regions of the potentials and how amplitudes and slopes are matched at the boundaries. A single step potential is chosen with random values for the left and right potential levels. For a randomly chosen energy level, which could be either above or below the right side of the potential step, the student is asked to give the analytic forms of the wave function at the left and right. For ease in typing, the student uses "a", "b", "c" and "d" to represent the general functions "Ae^{ikx}", "Be^{-ikx}", "Ce^{-Kx}", and "De^{Kx}" respectively. For example, when the student types "a+b", the computer writes "Ae^{ikx} + Be^{-ikx}" for him, and then proceeds to give a response appropriate to the answer. After calculating the wave numbers for the left and right regions, the student must attempt to satisfy the boundary conditions. When he has solved the equations and entered values for the real and imaginary parts of the coefficients, the computer plots the wave function to show what it looks like at the boundary. The student has an immediate check on his calculations and can see what to adjust if the amplitudes and slopes do not match at the boundary. This exercise gives the student visual assistance on solving this type of problem that he would not get on a typical homework assignment.

A recently developed lesson attempts to make the two-electron helium atom a possible area of study for undergraduate physics students. The goal for the student in this lesson is to find the ground-state wave function of one of the two electrons. This lesson has two major parts. In the first part, the student identifies the forces involved, derives an expression for the Coulomb potential using elementary electrostatics, then works through a simple analytic example of finding the Coulomb potential for a given spherically symmetric charge distribution. In the second part, the student is to find numerically a self-consistent ground-state wave function for the helium atom. The self-consistent calculation will be described in detail.

Below is the "index" page that the student keeps returning to during the numerical calculations. It shows the equations previously derived that must be solved, and gives a solution scheme with 6 basic steps describing a self-consistent numerical calculation. Many students never see such a calculation in their courses although the methods are often used in physics research.

$$\begin{aligned} \text{a) } & [(-\hbar^2/2m)\nabla^2 + U_C(r)] \phi(r) = E\phi(r) \\ \text{b) } & U_C(r) = -Ze^2/r + U_{e-e}(r) \\ \text{c) } & \nabla^2 U_{e-e}(r) = -4\pi e^2 |\phi(r)|^2. \end{aligned}$$

Solution scheme:

- 1) guess $\phi(r)$
- 2) solve (c) for $U_{e-e}(r)$
- 3) substitute $U_{e-e}(r)$ into (b)
- 4) find E and $\phi(r)$ from (a)
- 5) Compare $\phi(r)$ in (a) with initial guess
- 6) iterate until $\phi(r)$ is consistent

Choose a step (1-6): _____

Equation (a) is Schrodinger's Equation with the Coulomb potential $U_C(r)$; equation (b) gives $U_C(r)$ in terms of the potential of an electron in the field of $Z = 2$ protons and the electron-electron potential $U_{e-e}(r)$; equation (c) gives $U_{e-e}(r)$ in terms of the electron wave function $\phi(r)$. The student can attempt to solve these equations for $\phi(r)$ starting with any of the six steps, but he is told if his choice is inappropriate. For example, if he chooses step 4 before having done anything else, he receives the message, "... but I don't know the Hamiltonian yet. I need $U_C(r)$." Similar messages are given for other steps when enough information is not available to complete the step. The steps are detailed below. What appears on the student's terminal is boxed in. Quantities underlined indicate what the student types.

$\phi(r)$ will be assumed to be zero
past some maximum $r=r_{\text{max}}$.
Choose r_{max} : 2 Angstroms

The student enters a number for max. If the number is larger than 5, he is told that a smaller number would be more realistic. The initial guess to the wave function is then entered as a series of step functions. An example is shown below.

Guess $\phi(r)$.

$\phi(r)$ is to be entered numerically in step functions between $r=0$ and $r=r_{\text{max}}$.

$\phi(r) = 0, r > 2.000$ Angstroms.

$0. < r \leq .5$, $\phi(r) = 1.$

$.5 < r \leq 1.0$, $\phi(r) = .8$

$1.0 < r \leq 1.5$, $\phi(r) = .3$

$1.5 < r \leq 2.0$, $\phi(r) = .1$

The initial guess the student gives is plotted and normalized for him. The normalized values are available at any time later in the calculations by returning to the "index" page and requesting step 1.

Step 2: For each interval, give the numerical coefficients of the r^2 and $1/r$ terms in $U_{e-e}(r)$. Press DATA for a calculator, LAB for r and $\phi(r)$. $e^2=14.4$ eV-Å.
Press HELP to review the sample exercise; you can press ANS to fill in answers this time.

$1.500 < r \leq 2.000$, $\phi(r) = .050262$

$$U_{e-e}(r) = r^{-1} (13.18095 \quad) \\ + r^2 (-0.07619 \quad) \\ + \text{constant} (= .91429 \quad)$$

DATA, LAB, HELP, ANS refer to special keys on the student's keyset. Only one of the intervals is shown filled in here. The student must calculate the numbers for all intervals starting from larger r and working to $r = 0$. This is considerably simplified by being able to refer to the sample exercise for a spherical charge distribution worked out earlier in the first part of the lesson. A calculation mode is also available to the student.

Step 3: Accumulate the total Coulomb potential energy U_c . Give $U_c(r)$ as a function of r for each interval. Note: $Z=2$, $e^2=14.4$ eV-Å.
 $1.500 < r \leq 2.000$, $\phi(r) = .050262$
 $U_{e-e}(r) = (13.18095)/r + (-0.07619)r^2 + (.91429)$
 $U_c(r) = r^{-1} (13.18095-28.8 \quad) \\ + r^2 (-0.07619 \quad) + (.91429 \quad)$

This step is straight copying from step 2 except for the $1/r$ term which must now include $-Ze^2$ in the coefficient. In this case, the computer was left to do a subtraction before judging the answer as correct. The student is told whenever any of the answers are incorrect. When all numbers have been correctly calculated, U_c and U_{e-e} are plotted from $r = 0$ to $r = 1.5r_{\text{max}}$.

Step 4: The potential at the midpoint of 20 equal steps between $0 < r \leq 3.000$ Å is tabulated below. These numbers are used to find the new ϕ .

1) -364.39	11) -9.19
2) -108.73	12) -8.37
3) -58.21	13) -7.68
4) -37.30	14) -7.11
5) -26.27	15) -6.62
6) -19.80	16) -6.19
7) -15.87	17) -5.82
8) -13.38	18) -5.49
9) -11.56	19) -5.19
10) -10.21	20) -4.92

The potential will now be set up for plotting wave functions; ϕ is automatically started with value of 0 at $r=0$, as it must be if the equations are to remain finite with $1/r$ terms.

Try different energies until you find the lowest bound-state energy and wave function.

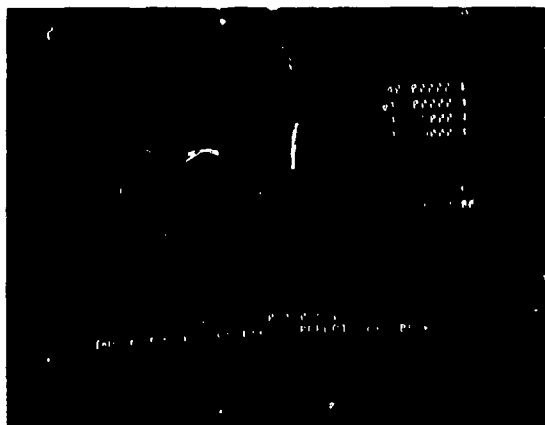
Press RACK when you are done.

At this point the student is transferred to a part of the lesson that plots wave functions for various potentials. The potential the student just calculated is set up between $r = 0$ and $r = 1.5r_{\max}$. Various energies are tried until the wave function shows the proper asymptotic behavior at large r . An illustration of this step is shown below in an actual photograph of a student terminal. The example shown does not yet represent a self-consistent wave function.

Step 5:

Here is the ϕ you just found, given numerically at 20 points between $r=0$ and $r=1.5r_{\max}$. ϕ is zero at $r=0$, and is normalized to unity.

r	$\phi(r)$	r	$\phi(r)$
1) 0.15,	...	11) 1.65,	...
2) 0.30,		12) 1.80,	
3) 0.45,		13) 1.95,	
4) 0.60,		14) 2.10,	
5) 0.75,		15) 2.25,	
6) 0.90,		16) 2.40,	
7) 1.05,		17) 2.55,	
8) 1.20,		18) 2.70,	
9) 1.35,		19) 2.85,	
10) 1.50,		20) 3.00,	



The student is given a table of values representing the wave function found in step 4. The initial and new wave functions are also plotted together for easy comparison. The student hopes that the two wave functions will be very similar and tries to improve the initial guess until good agreement is found. More steps could be used in the initial guess to make it smoother.

Step 6:

Enter a new guess for $\phi(r)$. I'll give you the potential to use for finding the new ϕ . (Press LAB if you want to change r_{\max} .)

$r_{\max} = 2.000$

$0 < r \leq$ _____ , $\phi(r) =$ _____

After the student has gone through all of the calculations once, he need not do again unless he wants to. Only a new guess to the wave function needs to be entered to start the search for a new bound state in step 4. The computer calculates the potential. Any of the intermediate results are available to the student by requesting the appropriate step number on the index page.

This presents a large portion of the lessons currently being used with physics courses in quantum mechanics at the University of Illinois. Some lessons attempt to bring to students knowledge and educational experiences not normally available to them because of complicated calculations or analysis involved. Others provide additional guidance, insight, and exercise in areas of particular difficulty. All seek to improve the quality of physics education.

ACKNOWLEDGEMENTS

Appreciation is extended to D. G. Ravenhall for originating the idea for the helium-atom lesson, and to B. A. Sherwood for helpful suggestions during the development and testing of this and other lessons. Thanks also go to D. K. Campbell, U. E. Kruse, M. B. Salamon, and H. J. Stapleton for their interest and cooperation in bringing computer-based education to their physics classes.

APPENDIX

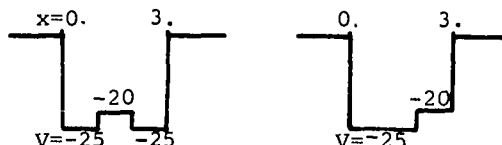
Below are a few of the homework problems assigned in the junior-senior level quantum mechanics course. They are to be worked using simulations of wave functions plotted by the computer.

Square Well

- Find the energy of the lowest-energy bound state ("ground state") of a symmetrical potential well of width 2\AA and depth -35 electron volts. Sketch the wave function plotted by the computer.
- Find the energy of the first excited state to within ± 0.5 eV.
- Change the depth of the well to -300 eV and confirm that the ground state energy is near the value expected for an infinitely deep square well.
- Obtain an estimate for the number of bound state for the -300 eV well. (Hint: Don't try to calculate all of them. Use what you know about the number of nodes in the bound-state wave function of energy E_n , i.e. the n th bound state of the system.

Asymmetric Well

Consider a simple symmetrical square-well potential of width 3\AA and depth -25 eV with a small potential "bump" in the bottom of the well. Two cases are shown, one with the bump placed at the center of the well, and one in which it is placed at one edge of the well.



- Find the ground state energy of the symmetrical well without the bump. Plot the probability distribution corresponding to this energy.
- Since the bump is relatively small compared to the well depth, an approximate value for the ground state energy in each case shown above could be found from the sum of the energy in the absence of the bump plus the expectation value of the extra potential energy represented by the bump. Predict qualitatively, and justify your prediction in terms of the probability distribution found in (a), whether the bump in the center has a larger, smaller, or equal effect on the ground state energy compared to the effect of the bump at the edge.
- Test your qualitative conclusion in part (b) by finding the actual ground state energies of the two wells with the bumps shown in the figures above.

Resonance Scattering

Consider an attractive potential of depth -20 eV and width 3.3\AA (square well). Study this potential well for plane waves incident from the left ($-\infty$) with positive energy E .

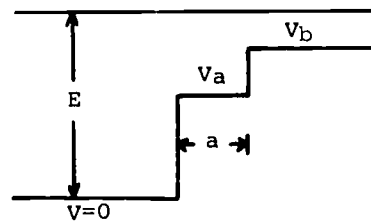
- a. Plot the transmission coefficient calculated by the computer as a function of energy in the range 5 eV to 15 eV. For what value of E do you find maximum transmission? What is the wavelength in the well?
- b. Compare your result with the analytic result found in the previous assignment.

Optical Coating

In optics it is possible to obtain 100% transmission into a medium of index n_3 from a medium of index n_1 by applying a very thin coat of another medium of index n_2 . This "trick" is used in coated lenses, and

$$n_2 = \sqrt{n_1 n_3} .$$

The thickness of medium 2 is a , where $a = \lambda_2/4$ and λ_2 is the wavelength of the light in medium 2. The quantum mechanical analog is shown below.



- a. Before going to the computer session, try to find a set of values for V_a , V_b , and E to satisfy the analog of $n_2 = \sqrt{n_1 n_3}$ and also try to estimate the appropriate thickness, a , for 100% transmission.
- b. With the computer, vary a to obtain optimum transmission, and make a plot of the wave functions for optimum transmission and of the transmission coefficient as a function of a around the optimum value.

INTRODUCTORY QUANTUM MECHANICS AND THE COMPUTER

John R. Merrill
Dartmouth College
Hanover, New Hampshire 03755
Telephone: (603) 646-2977

Introduction

Introductory quantum mechanics often is fairly abstract. The student is immersed in a melange of new functions and new analytic techniques; he is often not really at home with the mathematical background assumed by the course. He often feels all at sea. Moreover, the techniques of teaching quantum mechanics make what seem to the student strange divisions of material and strange choices of examples. On the one hand, he is told how important both the wavefunctions and energies for stationary states are, while on the other hand, the emphasis is on energies alone. The examples he meets are admitted to be very idealized because they are the only soluble cases. All-in-all the student gets a peculiar view of quantum mechanics as a highly abstract, very mathematical subject dealing with overly idealized physical situations.

When even a small computer is available, there is a wholly different approach to take. We have been using this approach with sophomores at Dartmouth, and the gain in the intuitive understanding of quantum mechanics is large. The computer approach uses a simple iterative technique, is general enough to cover most interesting three dimensional potentials, and easy enough to understand for the least sophisticated student. The method deals with energies, wavefunctions and probability densities simultaneously and on equal footing. It makes the existence of discrete levels very intuitive. It involves no sophisticated mathematics at all and yet leaves plenty of places where the analytic approach comes in naturally in a pre-motivated way.

This paper starts with a short discussion of one dimensional, stationary state, quantum mechanics at the sophomore level. It then moves on to discuss the three dimensional spherically symmetric potential. In upper level courses it is natural to follow the material discussed here with discussions of partial differential equations, and ways to solve the general three dimensional Schrodinger equation.

The General Method

For stationary state Schrodinger equation problems in one dimensional or spherically symmetric three dimensional problems, the wave equation can be written (in Hartree units):

$$\frac{d^2P}{dx^2} = 2(V(x) - E)P.$$

For the 3D problem $P(r) = rR(r)$ (where $R(r)$ is given in the separation

$$\psi = R(r)Y_l^m(\theta, \phi) \text{ and } V(x) = V_{\text{effective}}(r) = V(r) + \frac{l(l+1)}{2r^2}$$

The equation is completely general; any sensible potential can be considered.

The equation is solved in a simple way very much like an $F=ma$ problem. Figure 1 is a block sketch of the strategy (a simplified flow chart). First, you initialize the wavefunction P and its first derivative P' , and you choose the energy, E . Then (as the second step) you calculate the second derivative P'' at the point x using the Schrodinger equation. You take a small step Δx , and find the new P' and P'' and $P'_{\text{new}} = P'_{\text{old}} + P''\Delta x$, the new wavefunction P as $P_{\text{new}} = P_{\text{old}} + P'\Delta x$, and the new position as $x+\Delta x$. You print out the new position and wavefunction, and then repeat the process (from step 2) using the new x as the present point. Rather than leave the calculation as an open loop, you test each new x to see if it is still inside some preset region. At the end of the region, you see if the wavefunction is diverging; if the energy chosen determines an eigenstate, the wavefunction will not diverge.

As is often the case, convergence is worth worrying about, especially for potentials which vary rapidly in x . The simplest method for higher convergence is the initial $1/2$ step. This method is easy to motivate and works for many problems. An even higher convergence method is sometimes helpful. You can use fourth order Runge-Kutta or predictor-corrector methods.

There is an interesting higher convergence method which is both easy to motivate and easy to implement. This "fitting method" uses values of P'' at $(n+1)$ points in the (closed) interval $[x, x+\Delta x]$. These values are fitted to an n th order polynomial. You then write down the new P , and P' in terms of the integrated polynomial. If a parabolic fit is used (so that

$$P'' = A_0 + A_1\xi + A_2\xi^2 \text{ for } 0 < \xi < \Delta x \text{ in } [x, x+\Delta x] \text{ then}$$

$$P'_{\text{new}} = P'_{\text{old}} + A_0 \Delta x + \frac{A_1}{2} (\Delta x)^2 + \frac{A_2}{3} (\Delta x)^3$$

and

$$P_{\text{new}} = P_{\text{old}} + P'_{\text{old}} \Delta x + \frac{A_0}{2} (\Delta x)^2 + \frac{A_1}{6} (\Delta x)^3 + \frac{A_2}{12} (\Delta x)^4$$

This method is easy to explain to the student and easy to implement (just as the initial half step is). The method is also highly convergent: a method with an n th order polynomial fitted to the second derivative is related to an $(n+2)$ nd order Runge-Kutta. Finally, the method is very fast on most computers. Simple algebraic expressions are used for a quadratic fit to $P''(x)$, $P''(x + \frac{\Delta x}{2})$, and $P''(x + \Delta x)$, for example:

$$A_0 = P''(x)$$

$$A_2 = 4 * \left[\frac{A_0}{2} - P''\left(x + \frac{\Delta x}{2}\right) + \frac{P''(x + \Delta x)}{2} \right] / [(\Delta x)^2]$$

$$A_1 = 2 * \left[P''\left(x + \frac{\Delta x}{2}\right) - A_0 + \frac{A_2 * (\Delta x)^2}{4} \right] / \Delta x$$

This fitting method has been used successfully with freshmen and sophomores at Dartmouth.

One Dimensional Cases

The strategy of the sophomore course is as follows: We introduce the student to the usual history of quantum mechanics and to the de Broglie wave picture. We then introduce the one dimensional Schrodinger equation for stationary states and discuss the infinite square well and the finite step. These discussions motivate the concepts of discrete state, oscillatory wavefunction and decaying wavefunction. We sometimes discuss the finite well and barrier and state the result for the energies of the harmonic oscillator.

At this point we introduce the numerical procedure. We have the students apply the procedure to the finite well (whether or not we discussed that problem analytically). The students then apply the procedure to members of an interesting sequence of potentials $V(x) = |x|^m$. This sequence has the harmonic oscillator as one member; the harmonic oscillator is used as a check on the accuracy of the method. The rest of the sequence approximates an infinite square well with a rounded bottom. The student knows roughly what to expect for energies and wavefunctions from the simple analytic cases studied. Then the student turns to three dimensional cases.

All these one dimensional potentials are written symmetrically in x . We motivate and use the fact that the wavefunctions for symmetric cases are either even or odd in x (even or odd parity). The students then initialize their wavefunctions at the center of the well ($x=0$): $P=1$, $P'=0$ for even states; $P=0$, $P'=1$ for odd states. The program integrates the Schrodinger equation out to large x (well beyond the classical turning point), and the student finds out if the wavefunction diverges. The wavefunction diverges to $+\infty$ on one side of an eigenvalue and to $-\infty$ on the other. The student brackets eigenvalues quite quickly, and the sensitivity of the wavefunction's tail to energy is another indication of the meaning of discrete states. The number of loops of the wavefunction inside the well reminds the student of which state he has found.

One Dimensional Examples

One of the first cases studied is the finite well. The student has some feeling of what to expect from the short discussion of analytical methods. He starts with a deep well and follows states to shallower wells. Usually we ask that he watch the ground state (an even state), the first excited state (an odd state) and some other state which will be lost for the shallower wells.

Figure 2 shows the ground state wavefunction of a finite well with $V_0 = +100$. The wavefunction is displayed from the center of the well to beyond the edge of the well. The program can also give p^2 directly. The wavefunction is not normalized until the student calculates $I = \int P^2 dx$ and normalizes the central value for the calculation, $P(0)$, by $1/\sqrt{I}$. The energy is 1.07 (from the well bottom) in normalized units; the energy agrees with the analytic

solution to better than .1% even for reasonably large step sizes Δx . Changing the energy by $\pm 1/2\%$ makes the tail of the wavefunction diverge before the edge of the figure.

Figure 3 shows the first excited state, which has odd symmetry. Again the edge of the well is shown. The symmetry is made obvious to the student. The energy again agrees to better than .1%, and for a $1/2\%$ change in the energy ($E=4.26$) the wavefunction diverges before the edge of the picture.

Figure 4 shows the interesting set of potentials, $V(x) = |x|^m$. The harmonic oscillator, $V(x) = x^2$, is a member of the set, and its energies are known to be $E = (n+1/2)h\omega = (n+1/2)\sqrt{2}$ in these units. As is clear from the drawing, the potentials approximate the infinite square well out with rounded corners. By observing energies and wavefunction for this set of potentials as m increases, the student sees progressive effects of the increasingly broadened region where $V(x) > 0$. The wavefunction can decrease its kinetic energy by spreading out to fill the bottom of the well.

Figure 5 shows the ground states and second excited states (first excited even states) for $m=2, 10$ and 40 . The student sees that the qualitative shapes remain the same. As m increases, the wavefunction must tail off faster in the classically forbidden region. This tends to force the wavefunction away from the classically forbidden region because the logarithmic derivative is continuous at the edge.

Figure 6 shows the interesting effect on the energies of the two lowest even states as m increases. The minimum in E_1 near $m = 6$ is due to the competition between kinetic and potential energy terms.

Many other interesting potentials can be and have been studied. One is the harmonic oscillator with a Gaussian bump in the center. The states with energies of the order of the bump size are shifted in energy while higher lying states are left essentially unchanged. This potential also emphasizes that greater localization leads to higher energy.

A second interesting extension is to the continuum states. Choosing any energy in the continuum region leads to a wavefunction that does not diverge; the wavefunctions for continuum states become sinusoidal far from the effects of the potential.

Three Dimensional Cases

For spherically symmetric three dimensional potentials, the Schrodinger equation separates, and the θ and ϕ parts can be solved once and for all - numerically or analytically as seems best for the group of students. The most important part is the radial equation. This radial equation can be cast into a form exactly like that of a one-dimensional potential by the substitutions: $P(r) = rR(r)$ where R is the radial wavefunction. The potential is replaced by the effective

potential $(V(r) + \frac{\ell(\ell+1)}{2mr^2})$. The equation is solved in a way analogous to the one dimensional case discussed above. The only difference lies in the initialization. For three dimensional cases (or for non-symmetric one dimensional cases), the student initializes the wavefunction and its first derivative at large r . Since, for all interesting cases, $|V(r)| \rightarrow 0$ as $r \rightarrow \infty$ the asymptotic solutions (when $|V| \ll |E|$) have $p \propto e^{-\sqrt{2|E|r}}$ and $p' \propto -\sqrt{2|E|} e^{-\sqrt{2|E|r}}$

The student initializes the function at large r , chooses the energy and has the program integrate back toward $r = 0$ step-by-step. The student checks the wavefunction near $r = 0$. For $\ell > 0$, the wavefunction P must go to zero at $r = 0$. For $\ell = 0$, both P and $R = P/r$ must remain finite at $r = 0$.

Three Dimensional Examples

The strategy is completely general. The student can, if he wishes, use the numerical method on the hydrogen atom. The energies agree to better than .1%, and the radial wavefunctions, R , are indistinguishable from those of the analytic solution. Other interesting examples include the screened Coulomb potential and the 6-12 potential.

Figure 7 is a plot of the 6-12 potential $V(r) = 400(\frac{1}{r^{12}} - \frac{1}{r^6})$ and the energies of the lowest three $\ell = 0(s)$ states.

Figures 8, 9 and 10 show the wavefunctions $P = rR$ for these states. The energies are -66.2, -22.9, and -4.1. The horizontal scales of the figures are all the same so that the wavefunctions and the potential can be directly compared.

A second example is the screened Coulomb potential $V(r) = -\frac{e^{-r/10}}{r}$. Figure 11 shows

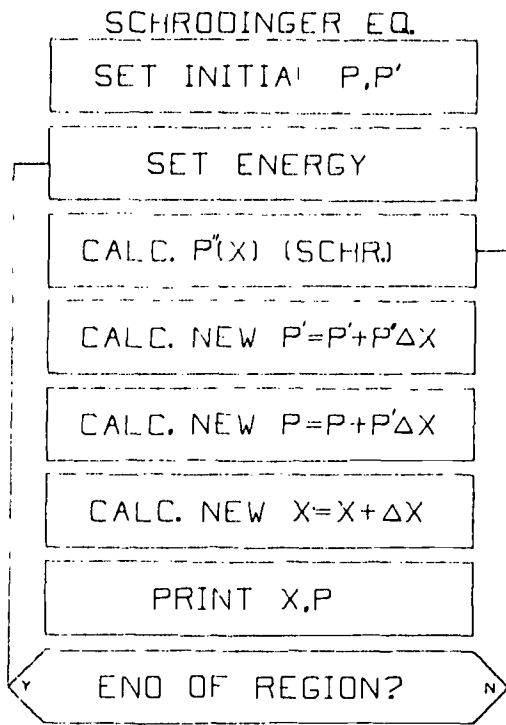


FIGURE 1. Block diagram of an iterative solution to Schrodinger's equation.

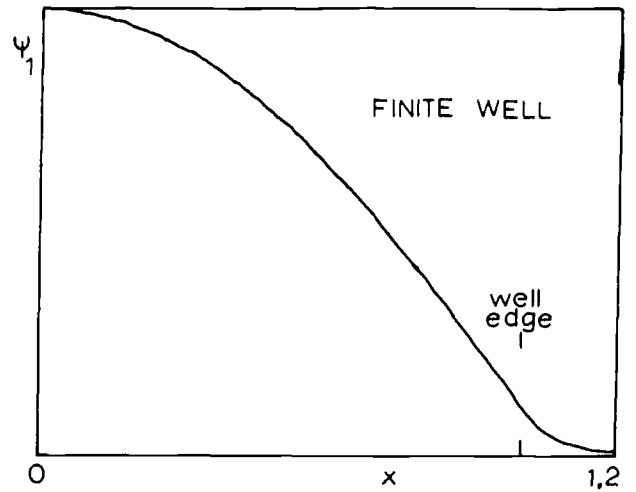


FIGURE 2. The ground state of the finite well. Only half the wavefunction is shown, namely that for $x > 0$.

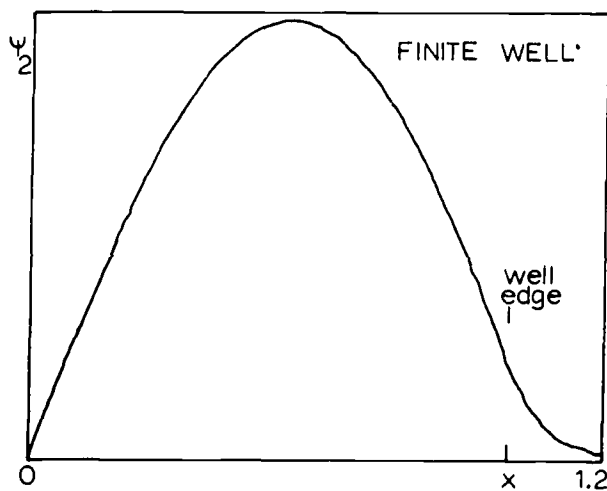


FIGURE 3. First excited state of the finite well.

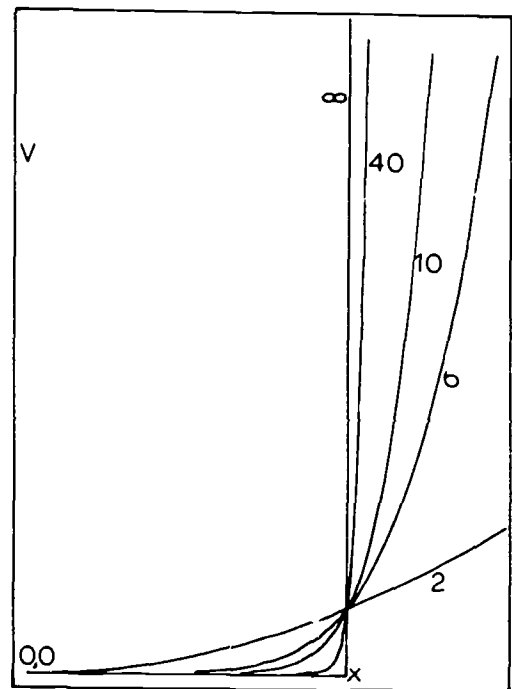


FIGURE 4. Members of the series of potentials $V(x) = x^m$ for $m=2, 6, 10, 40$ and infinity.

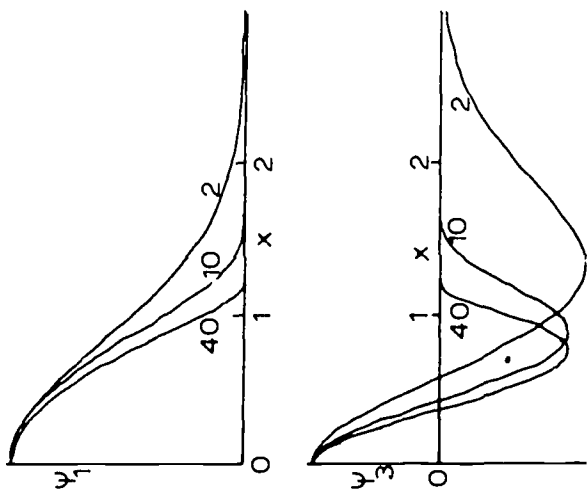


FIGURE 5. Ground states and second excited states for members of the potential $V(x)=x^m$.

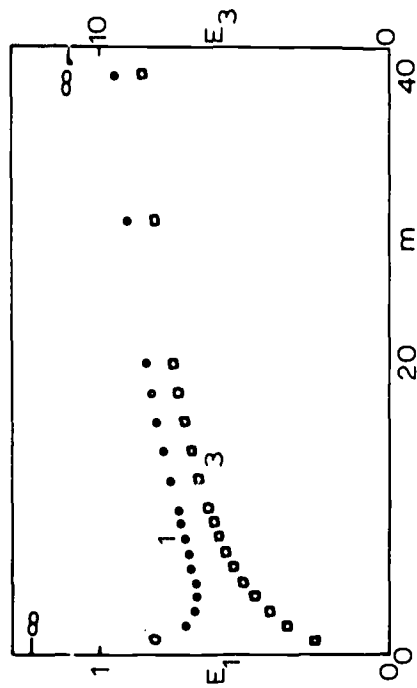


FIGURE 6. Energies of the ground state and second excited state for the potential $v(x)=x^m$ as functions of m .

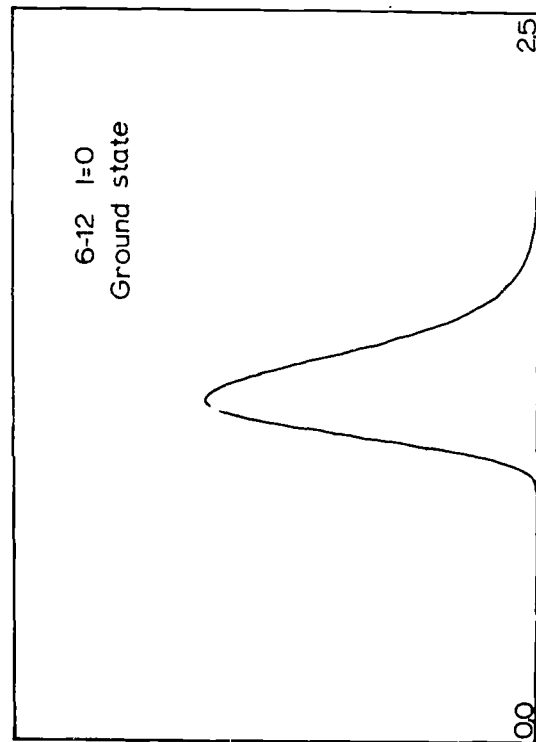


FIGURE 8. The $l=0$ ground state of the potential from FIGURE 7.

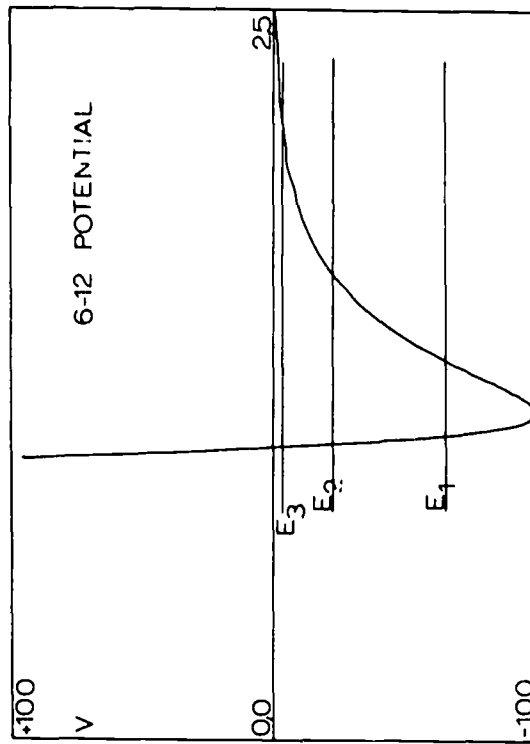


FIGURE 7. The 6-12 potential $V(r)=400(1/r^{12} - 1/r^6)$. The lowest three $l=0$ energies are also displayed.

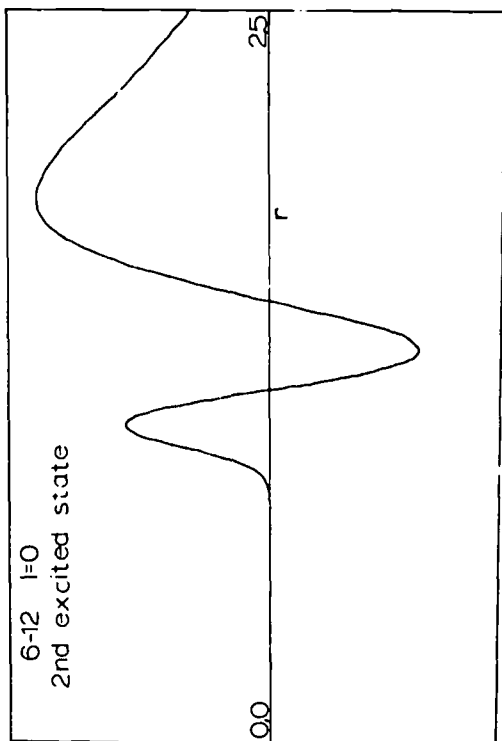


FIGURE 10. The second $l=0$ excited state for the 6-12 potential.

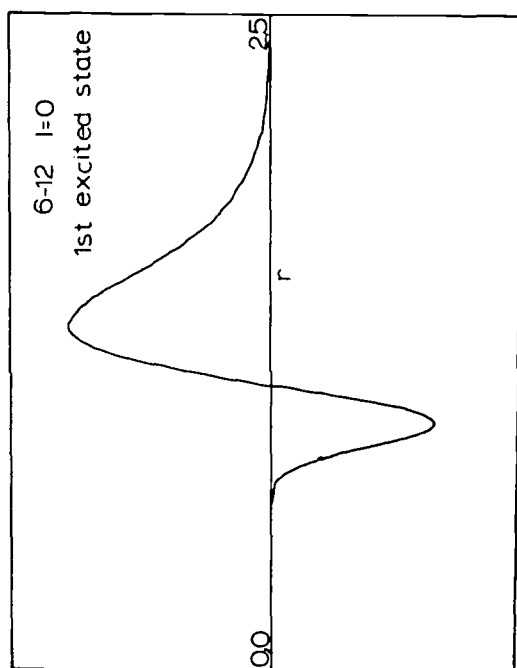


FIGURE 9. The first $l=0$ excited state for the 6-12 potential.

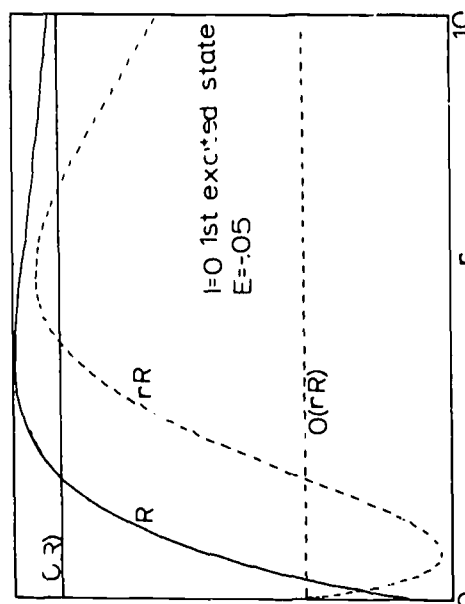


FIGURE 12. The first $l=0$ excited state of the screened Coulomb potential.

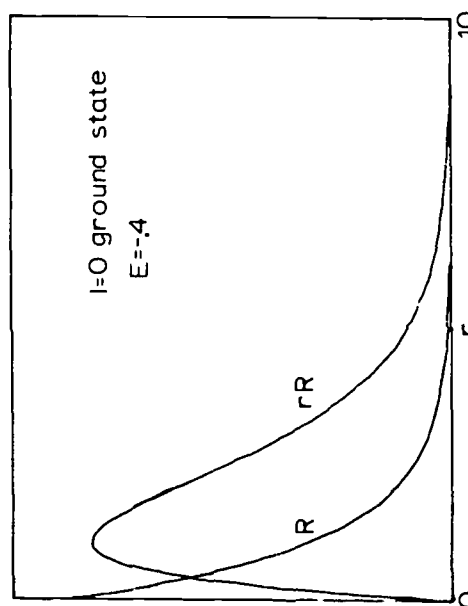


FIGURE 11. The radial wavefunction, R , (and $x=rR$) of the ground state of the screened Coulomb potential $V(r) = -\exp(-r/10)/r$.

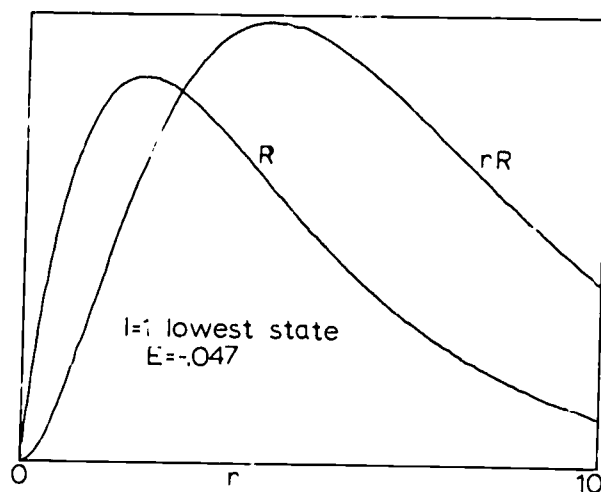


FIGURE 13. The lowest $l=1$ state of the screened Coulomb potential.

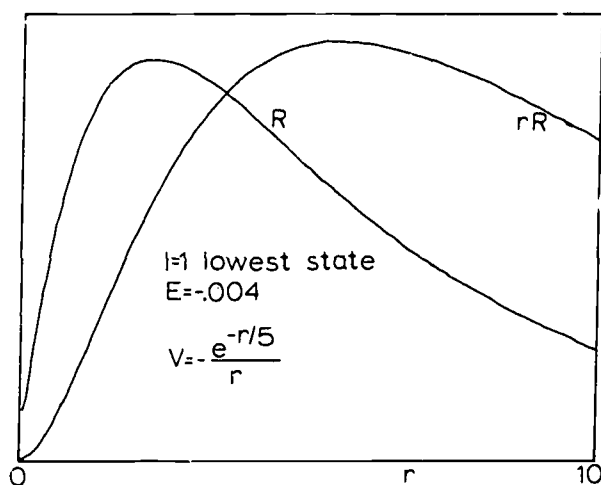


FIGURE 14. The lowest $l=1$ state of the (more tightly) screened Coulomb potential $V(r) = -\exp(-r/5)/r$.

381

R and $P = rR$ for the lowest energy s state ($l=0$). This ground state has an energy of $E = -.408$ (as compared to an unscreened Coulomb ground state of $-.5$ in these units). The wavefunctions, particularly for higher lying levels, are noticeably pushed towards $r = 0$ compared to unscreened wavefunctions. Figure 12 shows R and $P = rR$ for the first excited $l=0$ state; the energy is $E = -.050$ compared to an unscreened energy of $E = -.125$. For the unscreened Coulomb case, this state is degenerate with the lowest lying $l=1$ (p) state. Figure 13 shows R and P for this lowest p state. The energy for the screened case is $E = -.047$, so the state is not only shifted but the degeneracy is broken, too. This is one of the striking effects of breaking the strict $1/l$ dependence. A second striking effect is shown in Figure 14. This shows the lowest energy p state for $V(r) = -\frac{e^{-r/5}}{r}$ (tighter screening).

The state is just barely caught with an energy of $E = -.0043$. It is the only p state caught by the potential.

Conclusion

In upper level courses you can go from these examples to discuss partial differential equations and the solutions to the full Schrodinger equation. This is appropriate for a junior or senior level course. The discussion above has proved very effective at the sophomore level - the first introduction to quantum mechanics and the Schrodinger equation.

We have used this approach both in regular lecture courses and in self-paced (Keller Plan) courses. The system works well in both situations. The students gain considerable intuition in quantum mechanics. They can routinely sketch wavefunctions and guess approximate energies for states even in complicated potentials. This is a very useful application of computers to education.

INTERACTIVE CLASSROOM GRAPHICS*

Tim G. Kelley
Southern Oregon College
Ashland, Oregon 97520
Telephone: (503) 482-6400

David Grillot, Jeffrey D. Ballance, and Larry R. Hubble
Oregon State University
Corvallis, Oregon 97331
Telephone: (503) 754-1631 and 2490

Introduction

In previous papers (1,2) a pilot project was described which would allow an instructor to present computer-generated graphic information interactively in the classroom. In (1) the software requirements for such a system were discussed and in (2) the use of the system for production of visual aids (35 mm slides) was described. The software system, which has been named GROPE (Graphic Representation of Parameterized Expressions), is run on-line using the time-shared Oregon State University CDC-3300 computer. The system has as its underlying instructional philosophy the enhancement of the students' appreciation of various parameters which enter into a problem's formulation. Thus GROPE has been structured to develop and display parameterized families of curves. There are various methods available for defining these curves.

In the vast majority of cases, GROPE can be utilized effectively by a prospective teacher-user who has no previous computer experience or knowledge. While in the classroom the instructor issues simple mnemonic commands via a keyboard and the resulting graphic information is displayed on T.V. monitors or on an ordinary projection screen. Since an unlimited number of commands can be issued, displays may be continually developed, augmented and modified. Furthermore, since the computer normally responds to these commands in seconds, the timing and sequential structure of the displays may be controlled. A system which has these features provides a viable tool to supplement classroom lectures.

It is the purpose of this paper to describe: (1) the hardware configuration necessary for high quality classroom display, (2) the classroom use of GROPE in the Oregon State University Physics Department, (3) some examples of its capabilities.

Hardware

A hardware configuration has been achieved which allows GROPE to be used interactively in any size classroom. In addition to the time-shared computer, the hardware consists of the following equipment:

1. 4002A Tektronix Computer Graphics Display Terminal (Tekterminal)
2. 4501 Tektronix Scan Converter
3. A-912 ACP Kalart Tele-Beam or T.V. Monitors
4. Graphic Input Device (Joystick) *

The total cost of these items is roughly \$15,000 excluding communication costs. (With the advent of less expensive terminals this cost is reduced to roughly \$10,000.)

The Tekterminal, which provides the means of input and output, and the scan converter, which converts digital information to analog output, both reside on a cart which can be moved from one classroom to another. The Tele-Beam, which is also mobile, projects the analog information onto a projection screen so that graphic displays are easily viewable in large lecture rooms. In smaller classes (less than 15 students) the Tele-Beam is replaced by a TV Monitor. Since the equipment is easily moveable, the time required to set up the apparatus before a lecture is less than 5 minutes.

* This project was supported in part by NSF Grant GJ28453.

Figure I shows the arrangements of the equipment in an actual classroom use.



Classroom Use of Grope

Most of the faculty members who are using GROPE have had little or no experience with programming languages and computers. One of the primary considerations in designing the software was that such an uninitiated group be able to both learn and use the system in a minimal amount of time. The typical instructor spends less than one hour reading a manual and after two hours of assisted practice he is able to employ GROPE in his lectures even though he has acquired only a rudimentary knowledge of the capabilities of the system. More sophisticated use of the system develops rapidly as the instructor's need arises.

The response of faculty members using GROPE in courses at all levels has been enthusiastic. Large numbers of physically interesting problems which would be either impossible or difficult to analyze without the use of a computer are now routinely being examined in detail in various courses. The instructors find that not only do their students learn, but surprisingly they themselves gain new insights into the subject matter as they prepare their graphic presentations.

GROPE became classroom operational the last half of Fall quarter 1971. Since that time it has been effectively used in the following courses at Oregon State University:

<u>Course Name</u>	<u>Description</u>
Abridged General Physics	General Survey course for students with little mathematical background
General Physics	General Survey course for students having had college algebra and trigonometry
General Physics	General Survey course for freshman engineers
Physics I	Introductory physics for freshman physics students
Mechanics	Undergraduate mechanics course for junior and senior physics students
Modern Physics	Selected topics in modern physics for senior physics, engineering, and chemistry students
Dynamics	First year mechanics course for physics graduate and advanced undergraduate students

These courses were selected to initially test the feasibility of GROPE for classroom use, and efforts are now being directed toward expanding its use into other disciplines.

Capabilities and Examples

In order to create on-line classroom displays using GROPE, it is first necessary to define functions and then to specify the types of plots to be generated.

Functions may be defined in any of the following ways:

1. Typing analytic functional forms.
(These may have different forms in different domains.)
2. Typing coupled differential equation sets.
(See Appendix for the numerical method.)
3. Interfacing subroutines to GROPE.

It is also possible to define completely new functions in terms of one or more previously defined functions.

After the functions have been defined, they may be displayed in various ways. The following list exhibits some of the possibilities:

1. Cartesian or polar coordinates
2. Parametric or non-parametric plots
3. Multiple curves on the same axis
4. Two sets of axes on the same display
5. Powder plots of one or two dimensional distribution functions

The remainder of this section consists of examples which are intended to illustrate the instructional value of a system with these capabilities.

Example 1: The Simple Pendulum

The over-simplification required to achieve analytic solutions seriously handicaps the teaching of mathematical modelling for realistic systems. The resulting disagreement between model predictions and experimental reality is disconcerting to beginning students. Often they feel a disenchantment with mathematical analysis, since it seems incapable of dealing with "real life" problems. This is particularly true whenever the modelling is done through differential

equations. A more serious consequence is that students do not begin to learn the process of successive model refinement until late in their training.

As a simple example of how interactive graphics can help to alleviate these shortcomings, consider the one-dimensional pendulum problem. One begins by neglecting friction in the mounting, finite size of the pendulum bob, mass of the rod, rotation of the earth, etc.; in short, everything except the downward force of gravity. This is as it should be: the initial model should be kept as simple as possible. Even so, the resulting form of the equation of motion for the angular position of the bob,

$$x'' + \sin x = 0,$$

has no known analytic solutions.

Practicality often forces a serious pedagogical error at this point. For small angles, the replacement of $\sin x$ by x allows a solution to be obtained, but the validity of the approximation is far too difficult to evaluate at introductory levels. This is an unfortunate precedent to set, especially as it may be a student's first encounter with making approximations. A graphic presentation provides an excellent evaluation of the approximation, and can be understood and appreciated by students at all levels. Additionally, an interactive approach allows the analysis to take on the flavor of an investigation, with class participation.

The problem is initiated by typing the sequence (Input typed by the instructor is underlined.)

```

Y(T) = A * COS(T)
A = .2 BY .2
X''(T) = -SIN(X)
(Carriage Return)
X = .2 BY .2
X' = 0

```

which defines the harmonic solution $Y(T)$, and defines $X(T)$ through its differential equation and initial conditions. The value of Parameter (A) and initial conditions (X and X') are requested by the computer. Since we already know that the validity of the approximation depends upon smallness of angle, it would be most useful to generate a family of plots, parameterized by the amplitude of oscillation. Hence, A and X are established as automatically varying parameters by the above sequence. Following these definitions a series of commands must be entered to specify the functions to be plotted:

```

PLOT, Y
SPLIT, HORIZONTAL
PLOT, X.

```

After providing information requested by the computer concerning the plotting ranges, Figure 2 is obtained.



FIGURE 2



FIGURE 3

It should be emphasized that these input sequences are entered on-line and would completely define the problem to the computer. No prior preparation is necessary. In fact, it can all be done right in the classroom because it would take no longer than 30 seconds to enter all the necessary information and to get the first graph.

Each successive curve in the families is generated by depressing the carriage return key. Figure 3 might eventually be obtained, which shows very clearly the qualitative difference between X and Y: The period of the harmonic Y is independent of the amplitude, whereas the period of the solution X for the real pendulum increases with amplitude.

To obtain a more quantitative comparison both X and Y could be displayed on the same graph. The graph format can be respecified by a PLOT command without redefining the function, so Figure 4 can be produced in a matter of just a few seconds. A provision also exists for skipping over the range if the previous ranges are still satisfactory. For classroom use all shortcuts become important to minimize delays in the presentation. For this reason all command words (like PLOT, SPLIT, and HORIZONTAL) can be abbreviated by their first letter.



FIGURE 4

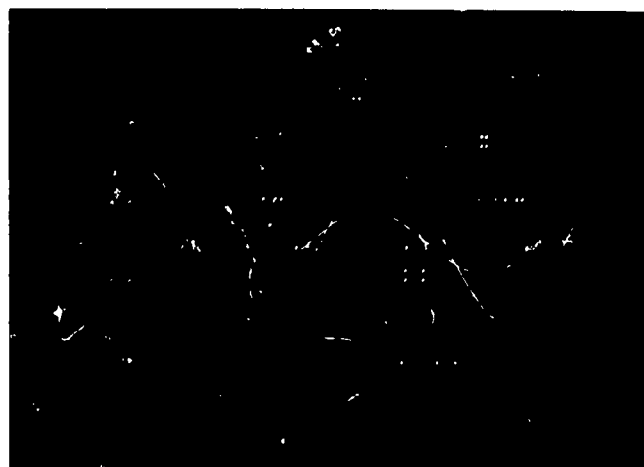


FIGURE 5

At this point the goal of obtaining a visual evaluation of the approximation has been accomplished. A good idea of the angular range of validity can be obtained, depending upon the accuracy required. It would be difficult not to notice, however, the striking resemblance of the real solution X to another cosine curve which is just frequency-shifted. This is an unexpected result that a curious student should want to investigate further. Why not "measure" the shifted frequency and compare X with a cosine curve of that frequency, as a further check? The coordinates of the appropriate points are requested by means of the "Del" command which employs the graphic input feature (see ref. 2), and within seconds the period of X is established to be 6.72 for the A=1 curve. Typing

```
PLOT,A*COS(2*PI*T/6.72)
```

then adds this curve to the display, producing Figure 5. The frequency-shifted cosine is truly a remarkable approximation to the real solution X.

Computer graphics has thus guided us (the class) naturally to an important conclusion which could not otherwise be reached except through a tedious laboratory exercise. Furthermore, the appropriate language for concisely formulating the evaluation criteria is now clear. A table, or perhaps a graph(+), of the frequency shift versus amplitude would provide a quantitative measure of validity. Figure 6 shows that this approach would be adequate up to an amplitude of about 2 radians.

- * This particular graph cannot be generated on-line without prior preparation. A subroutine would have to be written and interfaced to GROPE, which would allow it to be referenced in GROPE through the XTERNAL command. The interfacing amounts to placing the object deck of a function subroutine on magnetic disk file.

A slightly more advanced student might well be motivated to ask whether a correction so simple in nature cannot be deduced directly from the mathematics. Indeed there is a way, but a bit of insight is needed to get started. The first step is obvious, namely, use a better approximation for $\sin x$. To bolster confidence in the usefulness and validity of such a step, one might define $X_A(T)$ ("X-Approximate") through the differential equation

$$X_A''(T) = -X_A + X_A^3/6$$

and compare $X_A(T)$ with $X(T)$ for a large amplitude, as in Figure 7. The difficulty now is in treating this equation, which also has no known analytic solutions. Here is where an interactive graphics system can be used advantageously to assist and guide in sharpening analysis. The work thus far has suggested that X_A is well approximated by the form $\cos(\omega t)$, where ω is a shifted angular frequency. To establish this requires that the difference between X_A and $\cos(\omega t)$ be shown to be "small" in comparison to $\cos(\omega t)$. The standard method is to "guess" (by trial and error) an approximate form for the difference and then determine relative amplitudes by substitution into the differential equation for X_A . Since the trial and error process would require an hour or two for even good students, the guessed form seems to a class to be "pulled out of the air." But GROPE can readily generate $X_A(T)$ and plot it, so a natural attack is to plot the difference of $X_A(T)$ and $A \cdot \cos(\omega T)$, in the hope that it will have a simple enough form to recognize. (At this point the class or even the instructor may have no idea as to the size of this difference except that it is small. An automatic scaling feature wherein GROPE scans the data to determine appropriate axis limits for dependent variables is very helpful in such cases.)

The plot of the difference function shown in Figure 8 actually contains the needed information, but unfortunately it is not as recognizable as it might be. Because A and $X(0)$ have been chosen to be equal, the difference function as defined necessarily vanishes at $T=0$. This is arbitrary, and it is not unreasonable to hope that relaxing this restriction might yield a more recognizable form. Accordingly the function

$$Y(T) = (A + .001 \cdot B) \cdot \cos(2 \cdot \pi \cdot T / T_0)$$

was defined, and

$$D(T) = X_A - Y$$

plotted, with B varying. The scale of the variation in the cosine amplitude was suggested by the "size" of the function plotted in Figure 8. Figures 09 and 10 reveal the answer. At $B=6$ the difference function is very recognizable as $\cos(3\omega t)$, suggesting that

$$X_A = a \cos(\omega t) + b \cos(3\omega t)$$

be substituted, with a , b , and ω to be determined. When the analysis is completed (3) the mathematical support for the conclusion that $X = a \cos(\omega t)$ is indeed found, since b turns out to be very small in comparison with a . The result for ω is also in quantitative agreement with that obtained graphically.

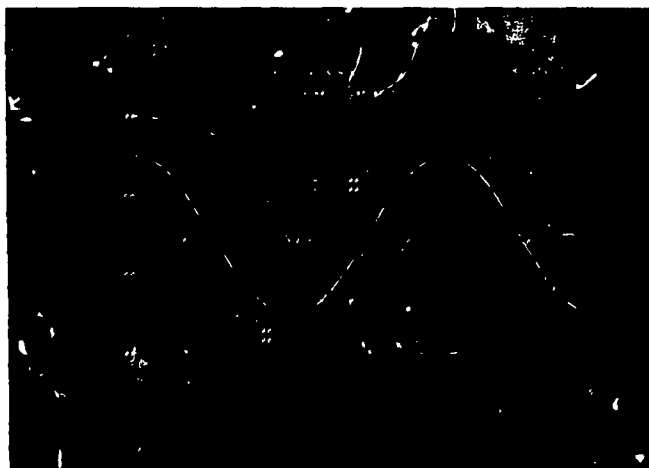


FIGURE 6



FIGURE 7



FIGURE 8



FIGURE 9

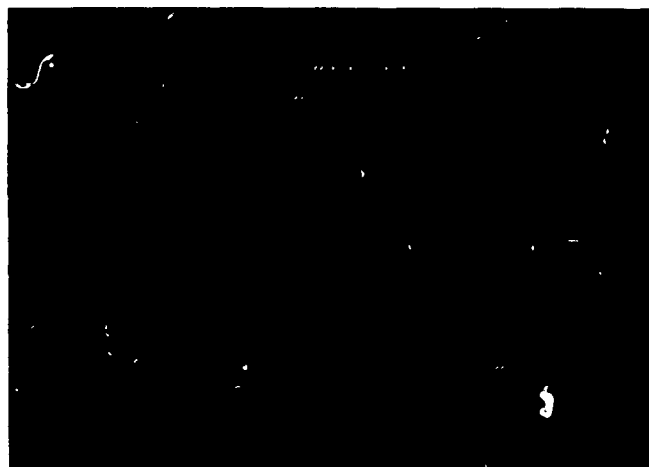


FIGURE 10

There are obviously a large variety of comparison curves which a student might now be inspired to request of the instructor. A real advantage of an interactive graphics system is the encouragement of curiosity. For example, we are now ready to successively refine our model by incorporating the frictional terms, etc., that were neglected earlier. Once a refinement is decided upon, the effect it has on the solution can be seen immediately.

Example 2: The Spherical Pendulum

One of the more pleasant surprises has been that instructors and students alike gain deeper insights into their subject material by the use of graphics. By being able to view "exact" solutions to problems under conditions where usual approximations fail, a new perspective is often acquired which leads to a more refined mathematical analysis.

A good example of this is the ideal spherical pendulum. It is just like the simple pendulum of the previous example, except that the bob is free to move anywhere on the surface of a sphere. Although solutions for special cases can be exhibited, the differential equations have in general no known analytic solutions.

The usual analysis(4) assumes the motion to be nearly circular, and yields a motion whose projection onto the horizontal plane is approximately that of a precessing ellipse as in Figures 11 and 12. The development of an expression for the precession rate is long and difficult for a student to follow, often because it is not clear where the analysis is leading. If a few accurate examples like Figure 12 for different rates of precession are shown to him, then he can see from the start that the salient features of this systematic motion ought to be extractable from the mathematics. Then when the approximate precession rate is obtained, it (1) is clear what it means, and (2) can be compared with that shown graphically.



FIGURE 11



FIGURE 12

The differential equation set could be specified to GROPE in whatever coordinates are convenient, but they are simplest in cartesian coordinates. The sequence.

$$\begin{aligned} X''(T) &= -X*F \\ Y''(T) &= -Y*F \\ Z''(T) &= -Z*F - G \\ F(T) &= [X'^2 + Y'^2 + Z'^2 - Z*G] / L^2 \end{aligned}$$

which would be followed by requests for the initial conditions and the parameter values for G and L would completely define the problem. Then the command

PLOT,Y(X)

produces the parametric plot of the motion projected onto the X-Y plane. Realism might be added by including frictional terms, $-B*X'$, $-B*Y'$, and $-B*Z'$, in the first three of the equations, yielding a result typified by Figure 13. Simple commands exist for printing function definitions and parameter values so the student need never wonder what functions are being displayed.

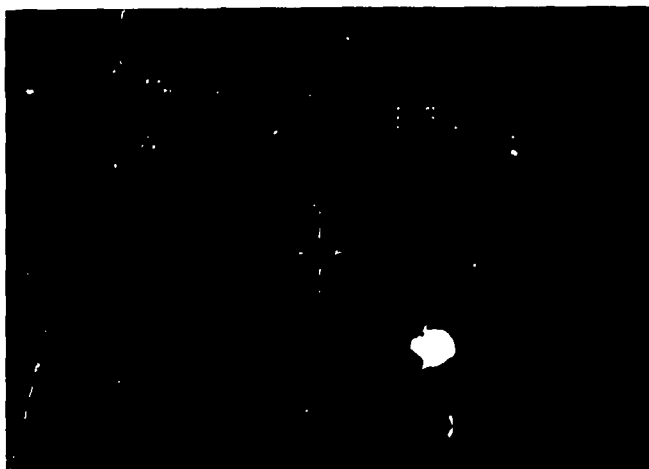


FIGURE 13



FIGURE 16

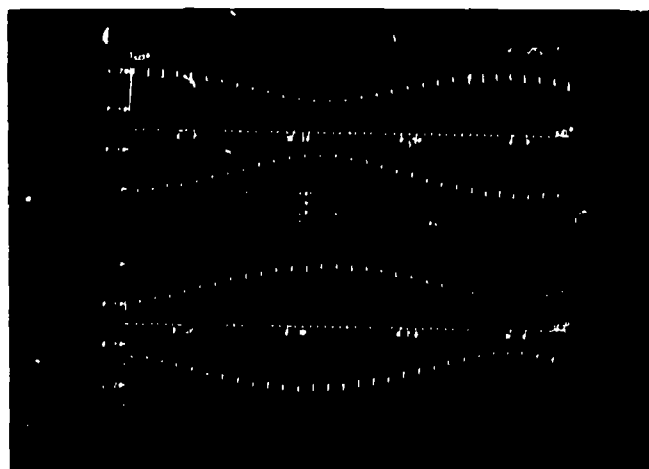


FIGURE 14

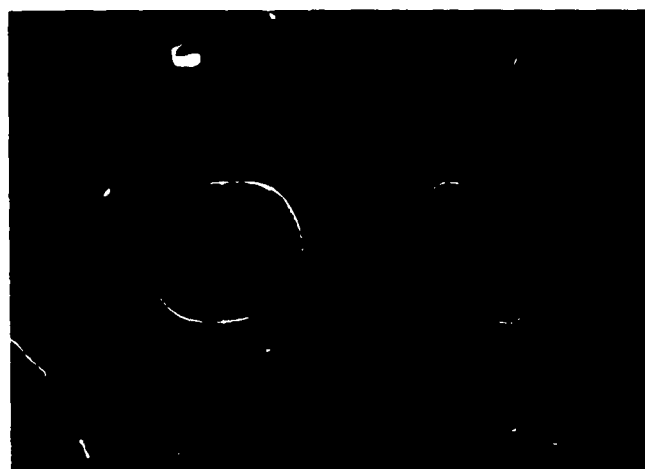


FIGURE 17



FIGURE 15



FIGURE 18

Returning to the frictionless case, increasing the energy (by increasing $I'(0)$) increases the size of the orbit until it is no longer "nearly circular," and the I-Y projection of it is no longer simple. One of the more powerful features of interactive graphics is the ease with which one can examine different aspects of a problem in search of a better perspective. There are a great many possibilities, of course, but it turns out that the most obvious ones lead to success. In particular, even though the X(T) and Y(T) motions are somewhat complicated, as shown in Figure 14, the Z(T) motion is amazingly harmonic at all energies. This is illustrated in Figure 15 which has graphs of Z(T) at three energies: a low energy, corresponding to the precessing ellipse; a high energy where the kinetic energy greatly exceeds the potential energy; and one in between. All appear very harmonic, although the intermediate energy curve contains considerable second harmonic component.

One of the pedagogical values of the spherical pendulum problem is the opportunity it provides to illustrate the use of conservation principles to extract pertinent features of the motion. Having thus been led to a view of the motion that is simple, these principles can now be used to understand why. If $V^2 = X'^2 + Y'^2 + Z'^2$ is eliminated from the equations of motion by conservation of energy, the Z" equation provides a full explanation of the harmonic nature of Z. Even though X" and Y" are not simple despite replacing Z by a harmonic form, the constraint equation yields a simple form for $p^2 \equiv X^2 + Y^2$. For large Z-oscillations the azimuthal motion can then be obtained by using conservation of angular momentum. The result of the analysis shows that the I-Y motion again approaches that of a precessing ellipse at high energies, but the precession is in a direction opposite to that at low energies. The exact solutions displayed graphically can then be used to verify these conclusions.

There are at the same time violent oscillations of the Z-coordinate. To help visualize the full motion, the projections in two perpendicular vertical planes can be displayed, as in Figures 16 through 18. Perhaps a better view of the motion is attained by typing in a few more equations to generate a 3-dimensional isometric representation of the motion, as in Figures 19 and 20. In all these cases the classroom presentation is far clearer than the pictures imply, because the dynamics of development are not lost. For example, the two curves of each of the split-plots of Figure 16 through 18 develop almost at the same time, so that the correlation between the two is clear.

All the curves shown were generated from functions defined on-line, including the isometric circles portraying the sphere. A total of 14 different functions had to be defined, and it is not practical to try to type that many equations in a classroom situation no matter how well-rehearsed the instructor is. To avoid undue delay when attempting complicated problems like this a FILE command is included for saving the definitions on a disk file. The FILE command can be given at any time and will save sufficient information to later restore the system to its present status. The GET command may be used later in class to retrieve this information. After the GET command is issued there are no restrictions in the use of GROPE and the instructor is free to utilize its full interactive capability.

Example 3: Schroedinger Equation- Powder Plots

Another area for potential use of computer graphics is in clarifying the meaning of functions of more than one variable. The present example is concerned with pictorially representing wavefunctions of the 3-dimensional Schroedinger equation. For a broad class of potentials(5), the wavefunction can be written as

$$\psi_{nlm}(r, \theta, \phi) = R_{nl}(r) P_{lm}(\theta) e^{im\phi}$$

where the angular functions, $P_{lm}(\theta)$, are the associated Legendre polynomials, and $U_{nl}(r) = rR_{nl}(r)$ satisfies the second order differential equation,

$$\left\{ \frac{d^2}{dr^2} + \frac{2m}{\hbar^2} \left[E - V(r) - \frac{l(l+1)}{r^2} \right] \right\} U(r) = 0$$

The potential chosen here is a Gaussian, $V(r) = V_0 e^{-r^2/a^2}$, with no known analytic solutions for the radial eigenfunctions.

The technique and instructional value of searching for the eigenvalue and eigenfunction which satisfy the radial equation and the appropriate boundary conditions have been discussed elsewhere(6,7). Figure 21 shows the final result of such an interactive trial and error search for the correct eigenvalue, E, and eigenfunction U_{13} . After $U_{13}(r)$ is determined, the instructor can easily display both the radial density $|R_{13}|^2$ and the angular density on the same screen as shown in Figures 22 and 24. As in Example 2, the connection of these 1-dimensional representations and a meaningful 3-dimensional picture of the physical reality is difficult to make without the aid of graphics. This connection can be made pictorially by generating a 2-dimensional powder plot representation(7) of the distribution functions, $|R_{13}|^2 |P_{30}|^2$ & $|R_{13}|^2 |P_{33}|^2$ as shown in Figures 23 and 25. These figures illustrate the

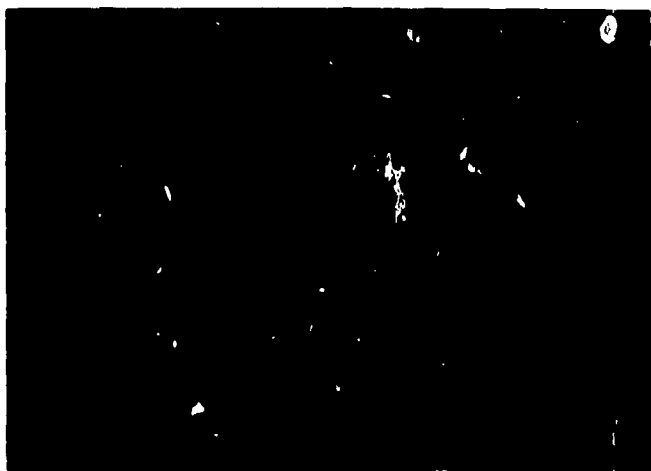


FIGURE 19



FIGURE 20

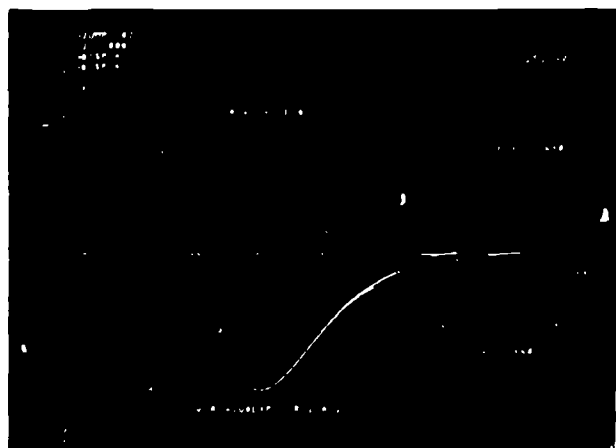


FIGURE 21



FIGURE 22

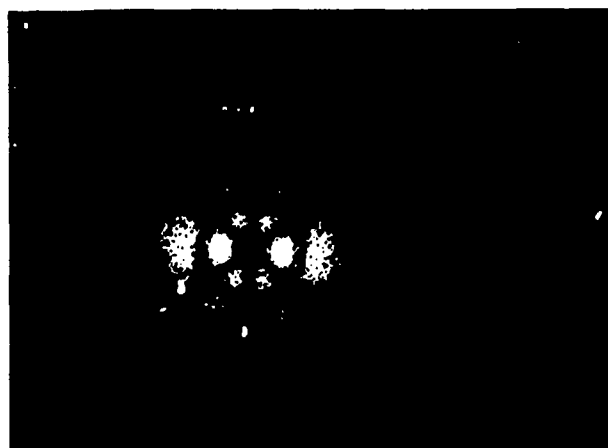


FIGURE 23



FIGURE 24

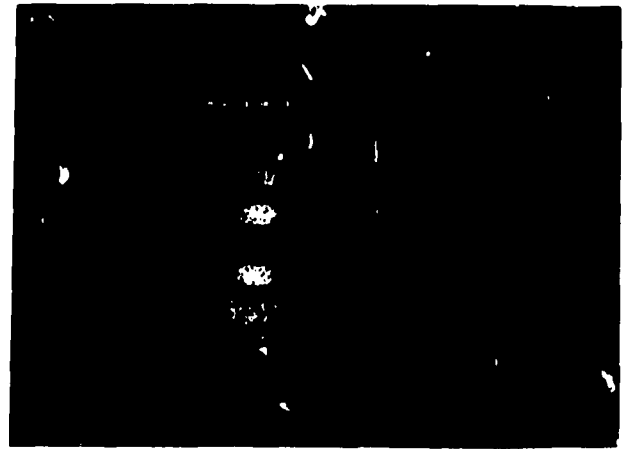


FIGURE 25

distribution function as a thin cross-sectional slab of uniform thickness which contains the polar-axis.

By using these figures and standard classical arguments the qualitative relationship between particle distribution and the projection of angular momentum can be clearly illustrated. For lower level non-technical students the powder plot provides a vivid method of introducing difficult elementary quantum mechanical concepts such as probability density.

The advantage of interactive graphics at this point is, as mentioned earlier, the flexibility and ease with which the instructor may pursue variations in this problem. He may choose to examine the effects of changing the strength or range of the potential on the particle distribution for the previous eigen states, or alternatively, he could choose to examine a completely different radial eigen state. These choices may very well be initiated in response to student questions. Figures 26 and 27, which represent the distribution function for an $M=0$, $L=3$, $M=0$ state, might be the result of such an extemporaneous classroom discussion.

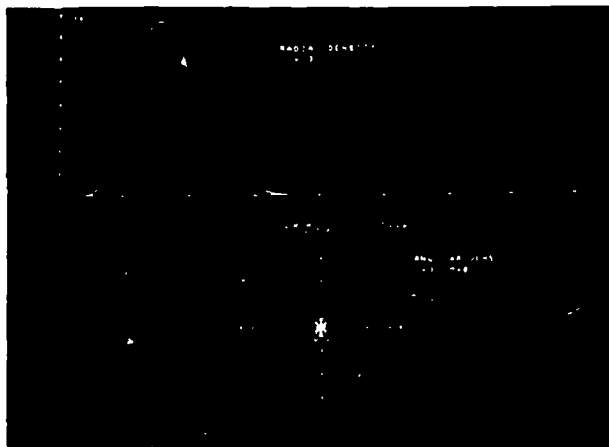


FIGURE 26



FIGURE 27

Example 4: Wave Packet Construction

The properties and behavior of localized wave packets are obscured by mathematical complexity. A graphic representation of the mathematics is very appropriate here, and the complexity makes computer generation the most practical, if not the only, way to achieve clarity.

Formally, a wave packet can conveniently be expressed as an integral

$$y_{k_0}(x,t) = \int_{-\infty}^{\infty} A(k - k_0) \cos(kx - \omega t) dk$$

where A must be a square integrable function. $A(k)$ is peaked at $k=0$ and is usually characterized by some measure of its width. The frequency ω is in general dependent upon k , and it is generally instructive to express ω as a power series in $(k - k_0)$. To acquire a full understanding of the packet construction and its time development, a student needs to appreciate how y_k depends upon the coefficients in this series, the width and shape of A , and the central component k_0 . The analysis of these characteristics could be achieved in the standard way using features of GROPE which have been discussed earlier. Consequently, we will not delve into these details here. Instead we wish to concentrate on the difficult and important concepts of phase and group velocity which often remain hazy for many students.

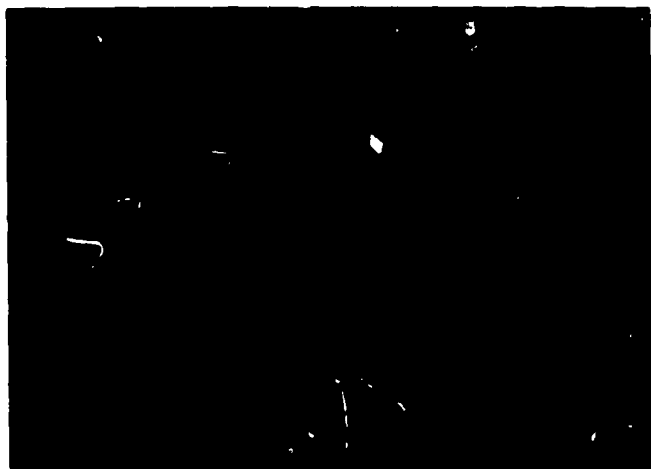


FIGURE 28



FIGURE 29

After $A(k)$ and $\omega(k)$ are defined, $Y(X)$ could be defined in two more steps,

$$F(k,X) = A(k - k_0) \cdot \cos(k \cdot X - \omega(k) \cdot T)$$

$$Y(X) = \text{INT}(F, L, U, N)$$

where INT is an internally defined function which numerically approximates $\int_L^U F(k,X) dk$ using N integration intervals.

Figures 28 and 29 show the propagation of a packet from $T=0$ (dashed curve) to $T=5$ (solid curve), and compares it with the propagation of the central component. To make the distance travelled apparent, the latter wave train was truncated outside a small region, as follows:

$$Z(X) = \cos[k_0 X - \omega(k_0) T] \text{ IF } -5\pi/2 < k_0 X - \omega(k_0) T < 5\pi/2 ; 0$$

Z then has the cosine form only if the argument of the cosine lies between $-5\pi/2$ and $5\pi/2$; otherwise it is zero.

These Figures show clearly the distinction between the phase velocity and the group velocity, the latter being equal to the coefficient of the linear term in the expansion for ω . The "spreading" of the packet, caused by the higher order terms in the expansion, is also apparent.

There are a number of physically interesting cases which could then be examined by introducing different forms for $W(K)$.

ACKNOWLEDGMENTS

The authors wish to thank the staff of the OSU Computer Center. We are particularly indebted to Dr. L. C. Hunter, Mrs. J. Baughman, and L. Ochs for their assistance, cooperation, and suggestions. Numerous members of the OSU Physics Department have made valuable suggestions in designing the system. Dr. P. R. Pontana and R. Srivastava have been particularly helpful and exceptionally patient in suffering through numerous revisions of the system. We also gratefully acknowledge an educational grant from Tektronix, Inc.

REFERENCES

1. Kelley, T. G., "On-Line Classroom Graphic Displays," in Proceedings of the Conference on Computers in Undergraduate Science Education, R. Blum, Editor, Commission on College Physics, 1971.
2. Kelley, T. G., "A General Graphics System for Computer Generation of Visual Aids," in Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, A. Luehrmann, Editor, Dartmouth College, 1971.
3. Kittel, Charles, W. D. Knight and M. A. Ruderman, Mechanics Berkeley Physics Course - Volume 1, McGraw-Hill, New York, 1965.
4. Fowles, Grant R., Analytical Methods, Holt, Rinehart, and Winston, Inc., New York, 1962.
5. Leighton, Robert B., Principles of Modern Physics, McGraw-Hill, New York, 1959.
6. Luehrmann, A., "Instruction Uses of the Computer," American Journal of Physics, March, 1967.
7. Ehrlich, R., "Physical Simulations for an On-Line Computer-Controlled Oscilloscope," in Proceedings of the Conference on Computers in Undergraduate Science Education, R. Blum, Editor, Commission on College Physics, 1971.
8. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill, New York, 1966.
9. Ralston, A. and Wilf, H. S., Mathematical Methods for Digital Computers, Wiley & Sons, New York, 1960.

APPENDIX

Solutions to the differential equations are computed using Adams-Bashforth-Moulton predictor-corrector formulas. The formulas have error terms $O(h^5)$ which are reduced to $O(h^6)$ by using the difference between the predicted and corrected values as a correction on each step. The difference between the predicted and corrected values is also used to control the local error. If this difference becomes too large the step width is halved and the calculations repeated using the new step size; if the difference is too small the step width is doubled and the calculations continued. Since the step width is allowed to change as needed and plotting and other calculations are done with a fixed increment, the solutions at the requested points are not always available. When this condition occurs the requested values are calculated using available values and an interpolation formula with error term $O(h^6)$. The starting values for the predictor-corrector method are computed using Runge-Kutta-Gill formulas. The first value is calculated using a stepwidth which is halved until the difference between the values computed using a given stepwidth and those computed using half that stepwidth is very small. A discussion of the methods used is given by Hamming(1) and by Ralston and Wilf(2).

COMPUTER GRAPHICS IN PHYSICS

Herbert Leckham
Gavilan College
Gilroy, California 95020
Telephone: (408) 842-8221

It is very useful to construct a picture of a phenomena, mathematical relationship, or process. The problem occurs when one attempts to do this. The diagrams drawn on a plane (a blackboard or sheet of paper) rarely look like the real thing. Therefore, it would be highly desirable to find a method to generate such pictures which would appear exactly as if a real physical representation were viewed with the eye. The digital computer furnishes such a capability when programmed to carry out a projective transformation. This paper will develop an approach to projective transformations and will examine several applications in physics. A BASIC program is given to carry out the projection.

The Idea Of A Projective Transformation

A number of ways exist by which an object in three space can be developed or projected upon a plane. The camera is probably the most common. Rays of light coming from the object are projected by the lens upon the projection plane which in this instance is the film.

The strategy which will be developed in this paper is quite different. The concept is portrayed in Figure 1. The process is described with respect to a cartesian coordinate system. The observation point is located somewhere in space with a line of sight directed towards the origin. Imagine that the object to be projected is described by a large number of points (x, y, z) one of which is shown in the diagram. If a ray is constructed which passes through the observation point and the point to be projected, the point of intersection with the projection plane can be found. If such a ray and corresponding intersection is constructed for each of the points on the object, the object itself can be developed on the projection plane.

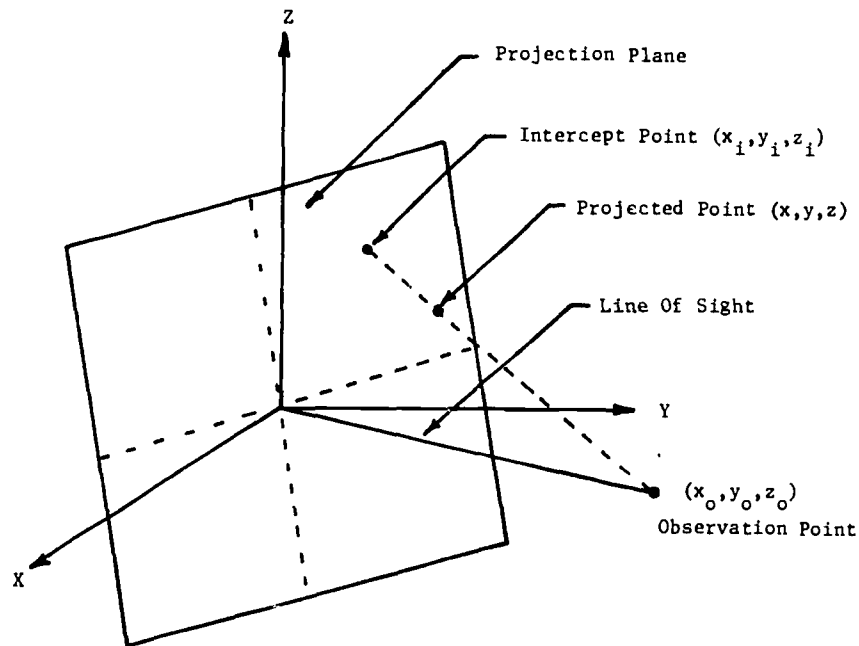


Figure 1

Mathematical Development of Intercept Point

From the mathematical point of view, when an observation point (x_0, y_0, z_0) and a projection point (x, y, z) are fixed, the intercept point (x_i, y_i, z_i) on the projection plane is uniquely defined. The problem is to locate the intercept point in terms of the coordinates of the observation and projection points.

The direction numbers of the line of sight joining the origin and the observation point are the coordinates of the observation point itself. Knowing these direction numbers, the equation of the projection plane (the plane passing through the origin and perpendicular to the line of sight) can be written. If this equation is written in terms of the variables (x_i, y_i, z_i) the result is

$$x_i x_0 + y_i y_0 + z_i z_0 = 0$$

The direction numbers of the projection ray passing through (x_0, y_0, z_0) and (x, y, z) are: $(x-x_0)$, $(y-y_0)$, and $(z-z_0)$. Note that for the purposes of each projection, the coordinates (x, y, z) are assumed to be constant. The equation of the projection ray which has the direction numbers above, and which passes through the point (x_0, y_0, z_0) is

$$\frac{x_i - x_0}{x - x_0} = \frac{y_i - y_0}{y - y_0} = \frac{z_i - z_0}{z - z_0} .$$

If (1) is solved for x_i the result is

$$x_i = -\frac{y_0}{x_0} y_i - \frac{z_0}{x_0} z_i .$$

But from (2), y_i and z_i can be written in terms of x_i . If these substitutions are made in (3), the resultant equation can be solved for x_i . Since there is nothing unique about the x coordinate of the intercept point, the other values y_i and z_i can be written from symmetry.

$$x_i = \frac{-(y_0^2 + z_0^2)x + (x_0 y_0)y + (x_0 z_0)z}{x_0(x - x_0) + y_0(y - y_0) + z_0(z - z_0)} ,$$

$$y_i = \frac{(x_0 y_0)x - (x_0^2 + z_0^2)y + (y_0 z_0)z}{x_0(x - x_0) + y_0(y - y_0) + z_0(z - z_0)} ,$$

$$z_i = \frac{(x_0 z_0)x + (y_0 z_0)y - (x_0^2 + y_0^2)z}{x_0(x - x_0) + y_0(y - y_0) + z_0(z - z_0)} .$$

Equations (4), (5), and (6) describe the intercept point. However, since the objective is to develop the point on a plane, a transformation must be written to transform the point in three space to a point in two space (the projection plane) where the point can be located by two coordinates.

Transformation of Intercept Point From E^3 to E^2

One way to carry out the desired transformation is to locate a pair of orthonormal basis vectors in the projection plane but which are still described in E^3 . If a vector is constructed

from the origin to the intercept point (also described in E^3) then the inner product of this vector with each of the basis vectors gives the coordinates of the intercept point in E^2 which can then be plotted.

An infinite number of sets of basis vectors for the projection plane can be found. The desired set is that one which does not result in a rotation of the object. One way to locate this particular basis set is to project the unit vector in the z direction $(0, 0, 1)$ onto the projection plane. Since $(0, 0, 1)$ corresponds to "up" with respect to the object, the projection of this point onto the projection plane will locate the proper "up" direction in the development of the object. Let the projection of $(0, 0, 1)$ onto the projection plane have coordinates (x', y', z') . The (4), (5), and (6) yield

$$x' = \frac{-x_0 z_0}{x_0^2 + y_0^2 + z_0^2 - z_0} ,$$

$$y' = \frac{-y_0 z_0}{x_0^2 + y_0^2 + z_0^2 - z_0} ,$$

$$z' = \frac{(x_0^2 + y_0^2)}{x_0^2 + y_0^2 + z_0^2 - z_0} .$$

Or, if

$$R_1 = \sqrt{x_0^2 + y_0^2 + z_0^2} ,$$

then

$$x' = \frac{-x_0 z_0}{R_1^2 - z_0} = V_1 ,$$

$$y' = \frac{-y_0 z_0}{R_1^2 - z_0} = V_2 ,$$

$$z' = \frac{(x_0^2 + y_0^2)}{R_1^2 - z_0} = V_3 .$$

Thus, if

$$R_2 = \sqrt{V_1^2 + V_2^2 + V_3^2} ,$$

then a unit vector in the "up" direction in the projection plane is

$$\vec{u}_2 = \frac{V_1}{R_2} \hat{x} + \frac{V_2}{R_2} \hat{y} + \frac{V_3}{R_2} \hat{z} ,$$

The unit vector from the origin towards the observation point is

$$\vec{u}_3 = \frac{x_o}{R_1} \hat{x} + \frac{y_o}{R_1} \hat{y} + \frac{z_o}{R_1} \hat{z}$$

where R_1 is defined by (7). \vec{u}_2 is one of the required set of basis vectors. If \vec{u}_1 is the other, it is defined by $\vec{u}_1 = \vec{u}_3 \times \vec{u}_2$, or

$$\vec{u}_1 = \frac{z_o V_2 - y_o V_3}{R_1 R_2} \hat{x} + \frac{x_o V_3 - z_o V_1}{R_1 R_2} \hat{y} + \frac{y_o V_1 - x_o V_2}{R_1 R_2} \hat{z}$$

Let

$$\vec{P} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \text{projected plotting coordinates,}$$

and

$$\vec{X} = x_1 \hat{x} + y_1 \hat{y} + z_1 \hat{z} = \text{intercept point.}$$

Then

$$P_1 = \vec{u}_1 \cdot \vec{X}, \text{ and } P_2 = \vec{u}_2 \cdot \vec{X}.$$

In matrix notation

$$P = TX$$

where

$$\begin{aligned} T_{11} &= \frac{z_o V_2 - y_o V_3}{R_1 R_2} & T_{12} &= \frac{x_o V_3 - z_o V_1}{R_1 R_2} & T_{13} &= \frac{y_o V_1 - x_o V_2}{R_1 R_2} \\ T_{21} &= \frac{V_1}{R_2} & T_{22} &= \frac{V_2}{R_2} & T_{23} &= \frac{V_3}{R_2} \end{aligned}$$

400

The Computer Program

Note that the projection divides readily into two sections. First, specification of (x_0, y_0, z_0) defines the transformation matrix T. This remains constant for a given problem. The second part of the projection involves computation of the intercept point. Since this depends upon the projected point (x, y, z) it must be done for every point to be projected. Consequently, this should be handled as a subroutine. The BASIC language program to carry out the projection is shown in Figure 2.

```

LIST
10 REM SET UP OBSERVATION POINT
12 PRINT "INPUT OBSERVATION POINT";
14 INPUT X0,Y0,Z0
16 REM COMPUTE TRANSFORMATION MATRIX T.
18 DIM T(2,3),P(2),I(3)
20 LET R1=SQR(X0^2+Y0^2+Z0^2)
22 LET V1=-X0*Z0/(R1^2-Z0)
24 LET V2=-Y0*Z0/(R1^2-Z0)
26 LET V3=(X0^2+Y0^2)/(R1^2-Z0)
28 LET R2=SQR(V1^2+V2^2+V3^2)
30 LET T(1,1)=(Z0+V2-Y0+V3)/(R1*R2)
32 LET T(1,2)=(X0+V3-Z0+V1)/(R1*R2)
34 LET T(1,3)=(Y0+V1-X0+V2)/(R1*R2)
36 LET T(2,1)=V1/R2
38 LET T(2,2)=V2/R2
40 LET T(2,3)=V3/R2
42 REM END OF COMPUTATION OF T.
44 GOTO 100
50 REM SUBROUTINE TO COMPUTE INTERCEPT
52 REM AND PROJECTED PLOTTING POINTS.
54 REM POINTS TO BE PROJECTED MUST
56 REM BE DESCRIBED BY (X,Y,Z).
58 LET R3=X0*(X-X0)+Y0*(Y-Y0)+Z0*(Z-Z0)
60 LET I(1)=(-(Y0^2+Z0^2)*X+X0*Y0+Y+X0*Z0+Z)/R3
62 LET I(2)=(X0+Y0*X-(X0^2+Z0^2)*Y+Y0*Z0+Z)/R3
64 LET I(3)=(X0+Z0*X+Y0+Z0*Y-(X0^2+Y0^2)*Z)/R3
66 MAT P=T*I
68 REM END OF PROJECTION SUBROUTINE. PROJECTED POINT
70 REM CORRESPONDING TO COMPUTED POINT(X,Y,Z) HAS PLOTTING
72 REM COORDINATES (P1,P2). IN CALLING PROGRAM ACCESS
74 REM PROJECTION SUBROUTINE BY "GOSUB 50." ALL SCALING
76 REM AND PLOT CONTROL MUST BE DONE IN CALLING PROGRAM.
78 REM VARIABLES (X0,Y0,Z0), MATRICES T,P,I MUST NOT BE
80 REM MODIFIED BY CALLING PROGRAM. LINE NUMBERS IN
82 REM CALLING PROGRAM SHOULD BEGIN AT 100.
84 RETURN

```

Lines 10 through 42 in the program accomplish the input of the observation point and computation of the transformation matrix T. Line 44 is an unconditional transfer to the first statement of the calling program which generates the points to be projected. Note that this requires that the first line in the calling program be 100. Lines 50 through 82 carry out the computation of the intercept point (designated by I_1 , I_2 , and I_3), and the matrix projection of this point from E^3 into E^2 . This subroutine is accessed from the calling program by GOSUB 50. The return to the calling program is in line 84. The result is the generation of a projected point (P_1, P_2) corresponding to each object point (x, y, z) . This projected point can be scaled and sent to a graphic display device such as an XY plotter or cathode ray tube. All scaling and plot control must be handled in the calling program.

Applications in Physics

In this section, several applications of the perspective projection algorithm will be presented. The examples have been chosen to show different and unusual graphic representations, and to give a feeling for the power of the algorithm.

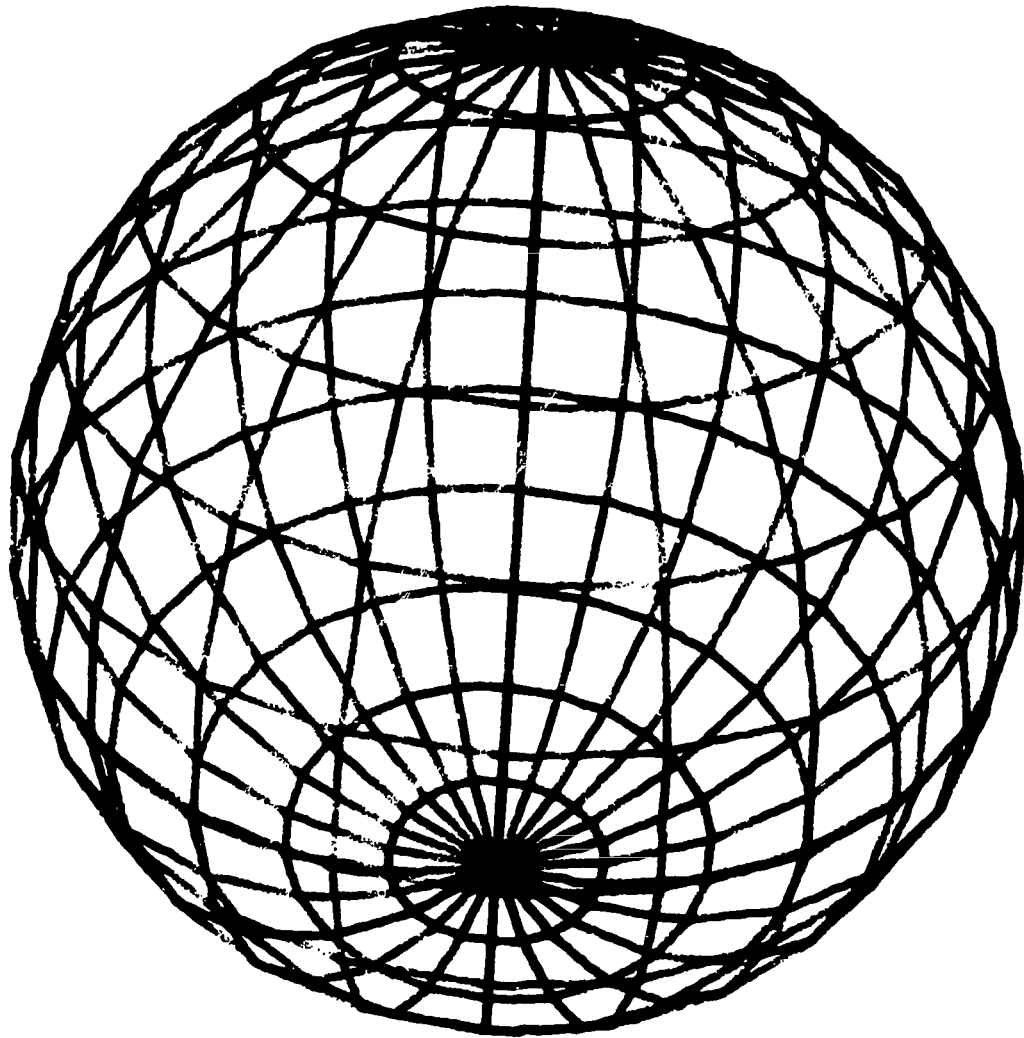


Figure 3 - Sphere

Figure 3 shows a sphere of unit radius centered on the origin as viewed from the point $(2, 2, 2)$. Latitude and longitude lines have been drawn every $\pi/12$ radians. Students often have difficulty visualizing a spherical coordinate system. However, an exercise to produce a drawing of this type has been found to be of great value. The program to produce the drawing (as well as the programs for the other examples) is in the appendix.

402

101

402

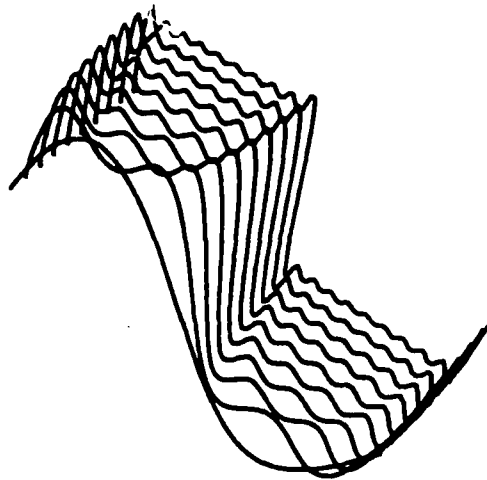


Figure 4 - Fourier Synthesis Of Square Wave

The Fourier synthesis of a square wave furnishes an interesting application for the projection algorithm. Figure 4 shows a drawing of such a synthesis as viewed from $(5, -5, 8)$. The square wave has a wavelength of 4. The first line in the drawing shows the first term in the Fourier series. Each subsequent line represents an additional term added to the series.

Figure 5 illustrates the potential field associated with an electric dipole. The charges are located below the plane in which the potential is computed to avoid the problem of infinite potentials. This example shows very clearly the perspective characteristic of the projection. The observation point is $(-15, 15, 10)$.

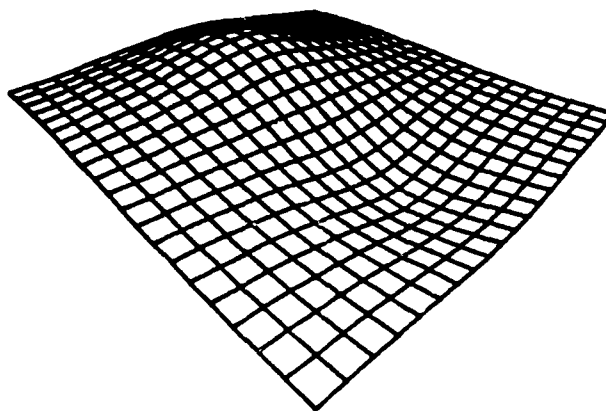


Figure 5 - Potential Field

403

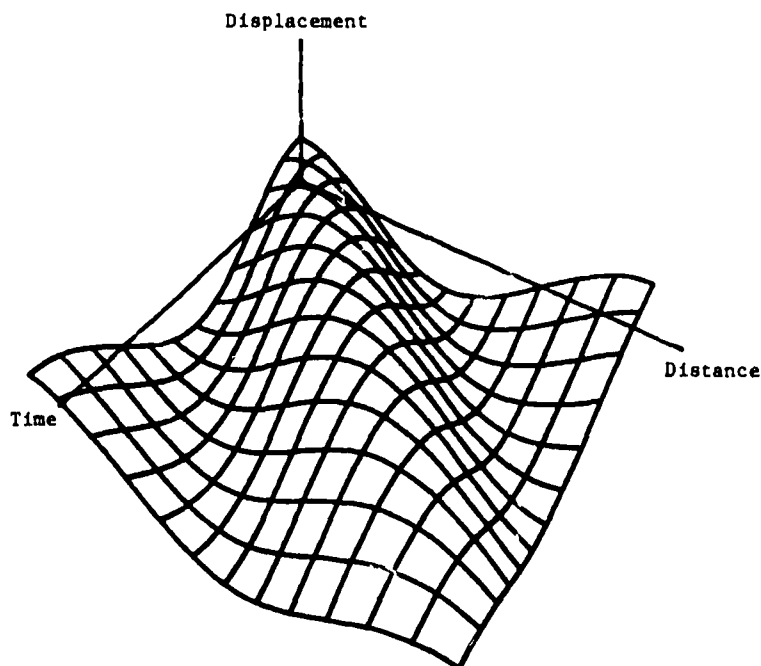


Figure 6 - Traveling Wave

A traveling wave with unit amplitude as viewed from (10,7,10) is shown in Figure 6. As time increases, the wave clearly moves to the right. Also, the drawing makes obvious the point that lines of constant time as well as lines of constant distance are sinusoids. This type of drawing can be used to illustrate the notion of wave velocity to great advantage.

Any of the drawings can be converted to stereo pairs by making an additional drawing with observation point chosen such that the new line of sight makes an angle of about 10 degrees with the old line of sight. If the two drawings are then viewed through a stereoscope, very interesting effects can be obtained. Students particularly seem to enjoy this type of exercise.

PROGRAMS

The programs utilized to generate the examples are contained in the appendix. While these programs are in BASIC, the structure of the algorithms is clear and it could be done equally well in any computer language. The programs in the appendix only generate a set of points (x,y,z) which then must be projected. Thus, the program in Figure 2 must be added at the beginning of each of the example programs.

The drawings in this paper were prepared on a Hewlett Packard 9100 XY plotter using a tape generated on Hewlett Packard 2007 Educational Computer System. The programs can be converted to an on-line mode by: deleting all CALL(1) statements, replace CALL(3) statements by PRINT "PLTL", replace CALL(2,A(1),2) statements with PRINT (A(1),A(2)), and replace CALL(5) statements with PRINT "PLTT."

Conclusion

The graphical techniques described in this paper have been used successfully with lower division students in physics and mathematics courses. Students can be taught to work in graphic projections whether they understand the mathematics of the projection or not. There are compelling reasons to get students involved in graphics if at all possible. Ideas which may be difficult to transmit emerge with clarity if approached graphically. It has also been observed that when students write programs to produce graphical representations they become completely submerged in the problem. This leads to a depth of understanding that is difficult to obtain by other methods.

APPENDIX

Program For Sphere

```
100 REM SPHERE
110 LET S1=3000
120 LET P=3.14159
130 CALL (1)
140 FOR A=P/12 TO 11*P/12 STEP P/12
150 CALL (3)
160 FOR B=0 TO 2.01*P STEP P/12
170 LET X=SIN(A)*COS(B)
180 LET Y=SIN(A)*SIN(B)
190 LET Z=COS(A)
200 GOSUB 500
210 NEXT B
220 CALL (5)
225 NEXT A
230 FOR B=0 TO 2.01*P STEP P/12
240 CALL (3)
250 FOR A=0 TO 1.01*P STEP P/12
260 LET X=SIN(A)*COS(B)
270 LET Y=SIN(A)*SIN(B)
280 LET Z=COS(A)
290 GOSUB 500
300 NEXT A
310 CALL (5)
320 NEXT B
330 CALL (1)
340 STOP
500 REM PLOT SUBROUTINE
510 GOSUB 50
520 LET A(1)=INT(S1*P(1)+5000)
530 LET A(2)=INT(S1*P(2)+5000)
540 CALL (2,A(1),2)
550 REM END SUBROUTINE
560 RETURN
999 END
```

Program For Fourier Synthesis Of Square Wave

```
100 REM FOURIER SYNTHESIS
110 LET S1=1500
120 LET P=3.14159
130 CALL (1)
140 FOR N=1 TO 10
150 CALL (3)
160 FOR X=0 TO 4 STEP 4.00000E-02
170 LET Z=0
180 FOR M=1 TO N
190 LET Z=Z+SIN((2*M-1)*P*X/2)/(2*M-1)
200 NEXT M
210 LET Z=Z*4/P
220 LET Y=2*N/10
230 GOSUB 400
240 NEXT X
250 CALL (5)
260 NEXT N
270 CALL (1)
280 STOP
400 REM PLOT SUBROUTINE
410 GOSUB 50
420 LET A(1)=INT(S1*P(1)+2000)
430 LET A(2)=INT(S1*P(2)+5000)
440 CALL (2,A(1),2)
450 REM END SUBROUTINE
460 RETURN
999 END
```

Program For Potential Field

```
LIST 100
100 REM POTENTIAL FIELD IN XY PLANE.
101 REM FROM UNIT POSITIVE CHARGE AT (5,0,-5)
102 REM AND UNIT NEGATIVE CHARGE AT (-5,0,-5)
110 LET S1=200
120 LET S2=20
130 CALL (1)
140 FOR X=-10 TO 10
150 CALL (3)
160 FOR Y=-10 TO 10
170 LET Z1=1/SQR((X-5)^2+Y^2+25)
180 LET Z2=-1/SQR((X+5)^2+Y^2+25)
190 LET Z=S2*(Z1+Z2)
200 GOSUB 50
210 LET A(1)=INT(S1+P(1)*5000)
220 LET A(2)=INT(S1+P(2)*5000)
230 CALL (2,A(1),2)
240 NEXT Y
250 CALL (5)
260 NEXT X
270 FOR Y=-10 TO 10
280 CALL (3)
290 FOR X=-10 TO 10
300 LET Z1=1/SQR((X-5)^2+Y^2+25)
310 LET Z2=-1/SQR((X+5)^2+Y^2+25)
320 LET Z=S2*(Z1+Z2)
330 GOSUB 50
340 LET A(1)=INT(S1+P(1)*5000)
350 LET A(2)=INT(S1+P(2)*5000)
360 CALL (2,A(1),2)
370 NEXT X
380 CALL (5)
390 NEXT Y
400 CALL (1)
999 END
```

Program For Traveling Wave

```
100 REM TRAVELING WAVE
110 LET S1=750
120 LET P=3.14159
130 CALL (1)
140 FOR X=0 TO 2.01+P STEP P/6
150 CALL (3)
160 FOR Y=0 TO 2.01+P STEP 2+P/50
170 LET Z=COS(X-Y)
180 GOSUB 500
190 NEXT Y
200 CALL (5)
210 NEXT X
220 FOR Y=0 TO 2.01+P STEP P/6
230 CALL (3)
240 FOR X=0 TO 2.01+P STEP 2+P/50
250 LET Z=COS(X-Y)
260 GOSUB 500
270 NEXT X
280 CALL (5)
290 NEXT Y
300 CALL (3)
310 LET Y=Z=0
320 FOR X=0 TO 7
330 GOSUB 500
340 NEXT X
350 CALL (5)
360 CALL (3)
370 LET X=Z=0
380 FOR Y=0 TO 7
390 GOSUB 500
```



```
400 NEXT Y
410 CALL (5)
415 CALL (3)
420 LET X=Y=0
430 FOR Z=C TO 3
440 GOSUB 500
450 NEXT Z
460 CALL (5)
470 CALL (1)
480 STOP
500 REM PLOT SUBROUTINE
510 GOSUB 50
520 LET A(1)=INT(S1*P(1)+4000)
530 LET A(2)=INT(S1*P(2)+7000)
540 CALL (2,A(1),2)
550 REM END SUBROUTINE
560 RETURN
999 END
```

COMPUTER GRAPHICS AND PHYSICS TEACHING

Alfred M. Bork and Richard Ballard
University of California, Irvine
Irvine, California, 92664
Telephone: (714) 833-6911

Last year at the Dartmouth Conference[1] we reported on the Physics Computer Development Project's use of computers in learning physics. We are concerned with procedures and areas in which the computer gives leverage in teaching difficult to obtain any other way. Our work has encompassed all modes of computer usage: dialog, computation, and simulation. Furthermore we have worked with full classes, testing our material with 150 students and rewriting it based on computer-saved feedback.

Our early work was with character-oriented terminals, either hardcopy or softcopy[2]. However, from the beginning we were interested in using graphic terminals; and recently our software development has permitted us to implement and use graphic teaching material for physics courses. This paper describes the types of usage of graphics in teaching we are exploring and the underlying software for graphics.

Pictures in Teaching

By looking in any textbook or visiting any lecture we can see that pictures, diagrams, and graphs are useful in teaching. The crude ability of alphanumeric terminals to present graphics has often been invoked in educational situations. Even a terminal such as the Model 33 Teletype can simulate point graphs by typing characters in fixed positions. Some of our dialogs produced crude diagrams and students often tried to graph output from their own programs.

Graphics in a timesharing environment has been expensive. However, new terminals and spectacular decreases in price promise that graphics will soon be widely available for students. These terminals can, under computer command, draw a line from one point to another on the screen and thus draw pictures of arbitrary complexity. Little use has been made of graphics with live students, so little is known about its effectiveness in teaching environments.

It is convenient to distinguish two types of computer usage in teaching, one in which the student writes his own programs and another in which the student interacts with existing programs. Implementation of graphic facilities is different in these two cases, and perhaps they even have different educational values.

Most of the work described here deals with this last type of usage. We will look first at the underlying graphic software used in dialog programs, then at comparisons of popular dialog programs which exist in both graphic and nongraphic forms. We next look at dialogs in which graphics is the key element. MOTION, a program for exploring classical mechanics, is given special attention. At the end of this paper we discuss briefly an interesting possibility for graphics within student written programs. As of this writing, the graphic software for this use is in early stages of implementation.

Underlying Graphic Software for Dialogs

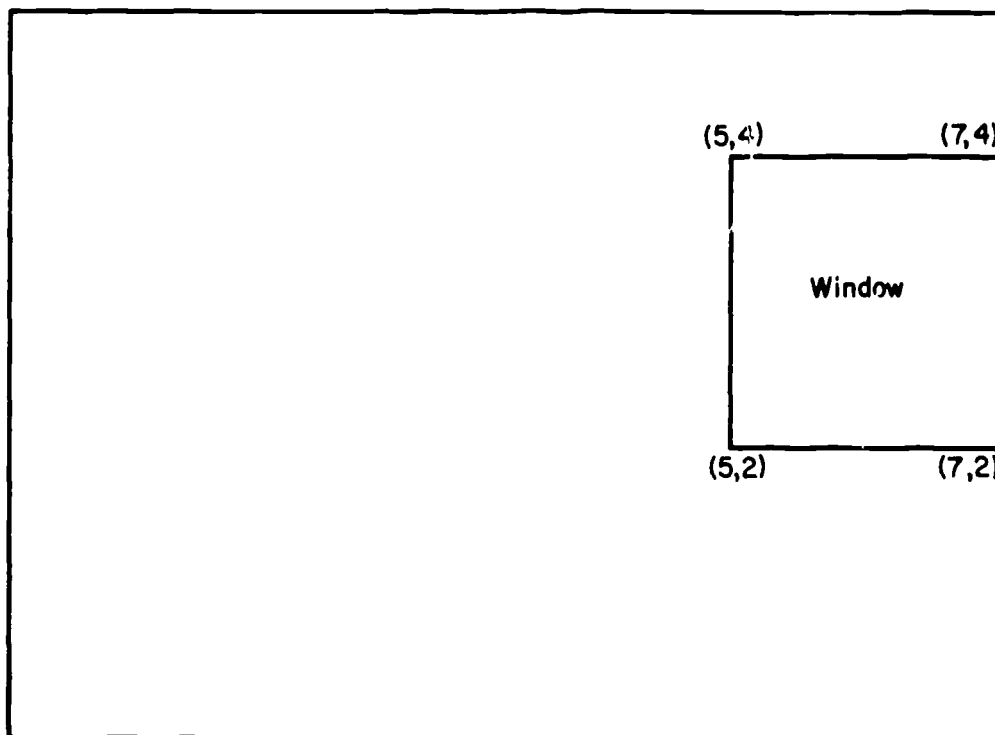
Our development of graphic software was an extension, within the same tradition, of our software for generating student-computer dialogs without graphics. Our approach has been to write assembly language macros which access assembly language subroutines. As teaching needs develop, we write new macros; so graphic teaching is a special case for our general tactic[3]. We make no attempt to give a complete software description; full documentation exists for those interested[4]. We illustrate some of the principal graphic macros, available under the BTM and UTS timesharing systems on the Xerox Sigma 7.

Graphic data can be computed within the program or fixed drawings can be part of the program. Typically in our programs the data is computed in code which originated as FORTRAN subroutines. The graphic data from the subroutine is in arrays.

The teacher might first decide where the picture is to appear on the screen. We might want to put several curves on at one time and to mix graphic with alphanumeric material. So the teacher needs an easy way of controlling where things appear. (In practice this is complicated by the fact that screens have different sizes and orientations; although our software covers this, we will not discuss it).

We let the user specify where he wants the curve drawn by a WINDOW command with specifications in inches from the lower left hand corner. Suppose, for example, that he wants a

curve, perhaps one of several, to appear in the box or window illustrated in the following diagram:



The statement within the program to establish this window would be:

```
WINDOW          (5,2) , (7,4)
```

The user can also specify a box around the window if he so desires:

```
WINDOW          (5,2) , (7,4) , BOX
```

Normally the box will not be drawn.

The teacher must next decide where within the window the curve is to be. Possibilities are numerous: we can choose to have the x and y or x, y and z data scaled so that it occupies the full window. Or the origin of the coordinate system can appear at the center of the window. Or the teacher may specify the coordinates of the ends of the window, again in two or three dimensions. The following uses of the macro SCALE let the user assign coordinates to the window:

```
SCALE           (X1,X2) , (Y1,Y2)
```

```
SCALE           (P,Q) , (X,Y) , (A,B)
```

P & Q are the minimum and maximum for the first variable plotted, X & Y for the second, and A & B for the third.

After establishing the window and the scale, we next draw the curve. The data will be in two or three arrays; the FORTRAN routine also returns us the number of points to be plotted. The command CURVE connects each of the "points" in the arrays with straight line segments. If the curve is three-dimensional, it is projected onto the two-dimensional screen.

Typical uses of the CURVE command are as follows:

CURVE	(X,Y,N)	2-D plot
CURVE	(HOR,VER,OUT,N)	3-D plot
CURVE	(MAX,(X,Y,N))	3-D plot, curve is scaled to touch window

on all sides.

CURVE (CENTER, (AA, BB, CC, M)) (0,0) is centered in the window.

The fourth command, the last we describe in detail, is AXES, drawing axes for two or three dimensional curves. Here are some examples:

```
AXES                                2-D axes using current scaling data
AXES (DIM, 3)                        3-D axes
AXES (MAX, (A, B, 50)), LIMITS
    Axes for largest possible
    curve. Maximum and minimum values
    of axes are shown.
AXES (LABELS, 'X', 'Y')
AXES (LABELS, 'VX', 'VY', 'VE')
```

Other macros are needed for positioning the beam, for erasure, and for specifying the graphic terminal; current software supports the ARDS 100, the Tektronix 4002, the Tektronix 4002A, and the Tektronix 4010. Adding a new terminal is a simple modification. A graphic program starts by asking the student which type of terminal he is using; with present terminals, unfortunately, the computer has no way of knowing the nature of the terminal. (We have information available[5], for those interested in the problem, about terminals for an educational environment.)

Graphic and Nongraphic Dialogs

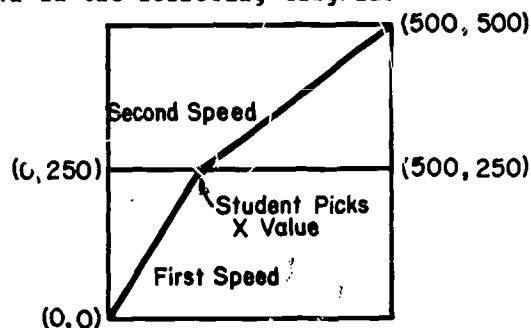
Several dialogs which use graphics are available in both graphic and nongraphic form; this gives us the opportunity to comment on the effectiveness of graphics. If a program exists in both forms, we would not expect it to be the same program, because the availability of new facilities indicates different possibilities in the teaching environment. Particularly in the early period of understanding the use and effect of graphics in learning it is valuable to have some similar programs attempting to exploit both graphic and nongraphic environments.

One dialog in both forms is the widely available one-dimensional lunar landing simulation. We have a nongraphic version, with numbers coming out, plotting the position in the usual typewriter way. This was written for our system by Noah Sherman and Steven Derenzo at the Lawrence Hall of Science, University of California, Berkeley.

The graphics lunar landing simulates a stylized spacecraft panel, with the position, velocity, and fuel indicated on graphs or gauges. The "instruments" are changed when the pilot gets close to the lunar surface, so that the student can get a more detailed view.

Both versions are popular with students; this is our most widely used program, both with physics students and others. It is clear from observing students that physics talent needed in one program differs from that needed in the other. In the "numbers" program, without graphic output, good students make " $1/2 at^2$ " calculations to determine fuel requirements in the final stages of landing; it was in connection with getting experience with the relations involving motion at constant acceleration that the dialog was developed. The graphics dialog makes students rely on curves of position and velocity vs. time and so has an entirely different "feel" to it. Students no longer make calculations but must develop an intuitive idea of what it is like to be involved in a constant acceleration environment with some fuel to slow down. We suspect, although we could not prove, that students learn more in the graphic environment in developing an intuitive feeling for the laws of motion. We contemplate tests involving these two versions.

A second dialog available in both graphic and non-graphic forms was developed by Murray Alexander of De Anza College in Cupertino, California. It is a "race", the Fermatopolis 500. In this somewhat unusual race the drivers have no control over the speed in the two laps. They go from (0,0) to (500,500) as shown in the following diagram:



A change in speed occurs only at $y = 250$. The speeds are announced in advance for each race for each of the two areas, and are the same for both "drivers." Each driver picks x corresponding to $y = 250$. The object is to win the race by keeping the time of travel as short as possible. The physicist sees that a minimal principle is involved, Fermat's principle in optics and leading to Hamilton's principle in mechanics. Variational principles are certainly an important way to formulate physical laws. Here is an important physical idea, vital in contemporary physics, almost totally neglected in the vast majority of introductory courses. Hence the computer has an opportunity to contribute significantly to learning in physics. Even a professional quickly discovers he is better off using his intuition than attempting to make the calculation, and the nonprofessional quickly sees what is involved in finding a minimum time in such a situation. The reward system has a higher payoff if the student's time is closer to the minimum; winning or losing depends on several races, with different speeds in the two regions.

With PERM it is more difficult to be definite about the advantages and disadvantages of the graphic versus the nongraphic form. In the graphic form you see the race happening, while in the nongraphic version you see only the resulting times; we believe that you get some feel as to why you win in the graphic version, because you observe that the winning person travels farther in the faster region. So there seems to be a gain in the graphics, but perhaps this gain is not sizable. Again we intend to do some testing with students using both versions, to see which form more quickly develops the students' intuitive understanding of action principles. We hope too to develop increasingly complex follow-up games of the same type, further extending the notion of variational principles.

GRAPH performs a utility function for students, one that we feel is very valuable for physics classes. As its name indicates, it graphs functions. Students should do some graphing by hand, but it is difficult to generate large numbers of graphs by hand. Yet seeing relations graphically is often an important part of understanding the physical aspects of a mathematical result. GRAPH makes it possible to examine many curves concurrently. Students enter the function in the usual notation with relatively few restrictions. The program contains its own parser which analyzes the functions and generate the graphic data. Students can request many different plots, set constants in equations to different values, etc. Functions are in parametric form, and plotting in two and three dimensions is available as in all of our graphic material.

This program has also found a use entirely different from that initially intended, in a "physics for artists" class. If we connect points on a curve which are not close together, we can construct beautiful patterns. The program allows students to control all variables, including the time step between successive points to be plotted.

MOTION

MOTION is a more elaborate instructional program using the computation, language recognition, and graphic facilities of the computer. In combination they have produced an exciting new tool and introduced several new teaching strategies.

MOTION was written to aid students and instructors at all levels in a study of equations of motion. The program offers each user a large repertoire of motions. Choices range from simple harmonic, central force, and constant acceleration to the very uncommon motions associated with two force centers or a multipole field. One can view the effects of anharmonicity, uniform electric and magnetic fields, or even the scattering from a nuclear force. A revised version will use parsing routines permitting students to write their own equations of motion.

Classical mechanics offers unique opportunities for using the computer to carry us far beyond present course boundaries. We have had ample demonstrations here and elsewhere that simple numerical methods, like the Euler method, can be taught to students at every level[8]. It is difficult to overstate the potential significance of introducing students so early to so powerful a tool. The beginning student can tackle problems and physical systems heretofore reserved to graduate students.

We are only just beginning to exploit the opportunities that a numerical approach provides. We have developed considerable experience in using these methods with large classes of students[1]. We have probed many of the pedagogical barriers to a wider use of computers. MOTION was written to overcome what seemed the more serious of these problems.

Perhaps the dominant objection to numerical solutions is that they produce a numerical result. For the most part it is difficult to interpret such results--to extract their physical consequences and go on to anticipate the solutions of other problems.

Wheeler proclaims as his First Moral Principle[9], "Never make a calculation until you know the answer." For many of our students, development of a physical intuition and an appreciation for the range of physical phenomena will serve them better than a knowledge of the mechanics for producing a particular solution.

With numerical solutions all results appear the same on the teletype, simple columns of numbers. Students rightly balk when asked to translate these numbers into graphs. They often view the process of changing parameters and replotting as tiresome busy work, yet repetitious plotting is the key element in learning from numerical solutions.

Using MOTION

MOTION uses a graphic dialog approach to overcome this objection. Students acquainted with the underlying algorithm can use that knowledge to explore the sensitivity of solutions to time step and other considerations. Other users, knowing nothing of such matters, will never encounter them. Using simple English they can choose motions for study, change constants and initial conditions, and then observe consequences of such changes.

The program can provide a form of instant experience to the student. In a very short time he can develop a qualitative understanding, for example, of any central force described by a power law or the sometimes spectacular orbits of a planet in a binary star system. The student is not restricted to plotting only spatial variables, nor in the choice of two- and three-dimensional projections. Virtually any physically meaningful variables can be plotted against any one or two other variables.

The explorative characteristics of an instructional program like MOTION are very important. Students bring to their physics classes a very narrow range of experience. While this has always been true, it has become increasingly acute as physics moves on to microscopic and macroscopic levels far removed from everyday observation. Students need this experience to understand physical principles. Great laws only appear as such when they help us to consolidate a variety of seemingly unrelated observations. MOTION offers a rich universe of examples. The unique behavior of total energy is nowhere more impressive than in three body motion. Its straight line time dependence stands out strikingly against the bizarre trajectories traced by other variables. We provide a wide range of physical examples, some obvious in their conservation of energy and in momentum, some not.

The sense of exploration in MOTION is quite real. If one ignores the infinity of variations produced by changing initial conditions and equation constants, over a hundred and fifty thousand distinctly different combinations of equations and variable projections are possible. Most of these have never been seen before by anyone. As a consequence, every user has the opportunity to learn something new and make genuine discoveries. Unlike most instructional programs, both the instructor and the student users are offered an opportunity to learn. If anything, the instructor's knowledge and experience may permit him to learn even more than the student. This last aspect has been exceedingly important in gaining faculty acceptance for the program. Instructors can test the effectiveness of the program on themselves.

Response Recognition in MOTION

Explorative programs like MOTION are difficult to program. The bulk of the computation and display options are straightforward; the tricky question is how to educate the user in the existence and operation of so vast a collection of options: all the equations, variables, projections, scaling, families, 3-D aids - like rotation and dashing. We chose a dialog strategy that puts the student in control of the program flow, letting him call for the facilities he wants. We take full advantage of dialog technique as a means of producing stand alone programs. It requires no prior instruction or descriptive handouts and adapts to the terminology and abilities of an enormous range of users.

MOTION attempts to recognize any question or request relating to its functions of selecting, solving, displaying motion. Although this sounds to be most difficult, it is not an impossible task. It departs completely from the patterns of program flow found in computer dialogs on programmed instruction. Students accustomed to these conventional dialogs will sometimes ask, "Where am I in the program? What can I do next?" The answer is that despite appearances they are always at the same point and that they are free to try anything they want.

Each input is inserted into a ring which performs an exhaustive search for key word fragments or symbols. As successive requests or questions are often related, the search process is made most efficient by inserting the input adjacent to the last successful key match. Suppose the student's last input was recognized as assigning a new value to one of the initial conditions. He is more likely to change another or to ask for a "plot" than to request another equation of motion. The program will do either, but checks first for the most logically related.

Once the presence of one or more key word fragments or symbols has been detected, subroutines are called to break down the message syntax. If parts are missing, the program requests their entry. Here again we try to avoid any "flow traps." We look first for the missing

information; if not present, we reinsert the new input into the test ring, on the chance that the user has disregarded our question and changed the subject.

These recognition facilities have proven quite effective. Anyone who knows what he wants, can ask for it. If clearly stated, he will usually get it. He need change only those things he wishes changed; all others will remain the same. If he does not set equation constants or initial conditions before asking for a plot, the program loads in an interesting example and proceeds to plot it.

MOTION employs several techniques for enlarging the student's knowledge of its facilities. The imaginative user will ask questions. This does not come easily; many users put off "wasting their time" until a high level of responsiveness has been demonstrated. Some reject the notion entirely. This may be the result of a previous exposure to computers; the totally uninitiated often accept the idea with great glee and begin asking questions having little relationship to the program.

Questions are usually answered with an excess of information. The requested facility is described and notice is taken of other related facilities perhaps unknown to the user.

Students having trouble with the program are also given an opportunity to learn. Whenever an input cannot be recognized, failing all tests, the program randomly selects a message appropriate to that area of the program. It describes some of the available features, using quotes to emphasize recognized terminology.

Observations on Student Use of MOTION

MOTION has been used by a spectrum of students, instructors, and computer professionals. It is, first of all, highly popular. Left to themselves, many students have spent the better part of a day running the program and return frequently thereafter. It was immediately adopted, highly recommended, and heavily used in the upper division mechanics course. Previously, the instructor had seen little use for computers in teaching. We are now testing the program with large classes at the introductory level.

The ways of using MOTION are varied. Instructors could connect the graphic terminal to a scan-converter and use the program as a televised demonstration in lectures. The format free, natural language approach in communications makes it very easy for instructors to learn its use; the absence of flow keeps them from wasting class time, if an error is made.

It is most often used by students as an adjunctive aid, available at any time. Students move easily between physical systems and discover quickly both two- and three-dimensional projections. Their reaction to the variety of variables that can be plotted seems to depend upon their educational level. The beginning student looks only at plots of position and time coordinates. The odds are against his discovering anything about energy conservation, momentum, angular momentum or motion in phase space until he has received some formal instruction. Our observations seem supportive of Bunderson's findings that directed instruction is more efficient than a pure discovery approach for the average or below average student [9]. In its present form MOTION serves these students by offering them a universe of dissimilar motions in which to test their newly learned abstractions.

Graphics in Student Programming

Our computer usage with students has involved students interacting with canned programs such as those just described, graphic and non-graphic, and it has also demanded that students write their own programs for solving physics problems. In the beginning course the two usages are about equal; in both cases comments of students at the end of the course suggest that we are at a reasonable level with regard to the amount of usage, although our students "vote" more favorably for dialogs than for computation. The total computer usage in the beginning course is about an hour and a half a student each week.

As of this writing we provide no graphic facilities which average students can use in their own programs. Knowledgeable students could use our general graphic software, but this demands more knowledge than we can reasonably expect from many beginning physics students. Many students resort to character-type plotting.

Data to be graphed in physics programs is primarily array data. X and Y or X and Y and Z coordinates are calculated for many, many points, and then the resulting arrays are converted into lines on the screen. Using graphics naturally within a student-written program depends on the ability of the programming language to conveniently construct and manipulate arrays of data. Two existing interactive graphics systems, the Culler-Fried and the Harvard TACT systems, both

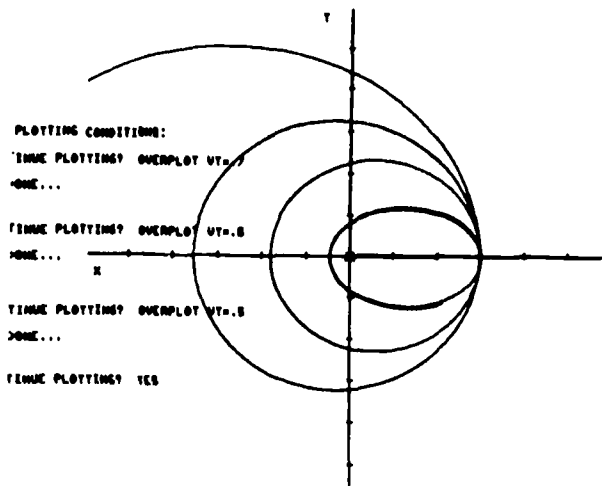


Figure 1

Figure 2

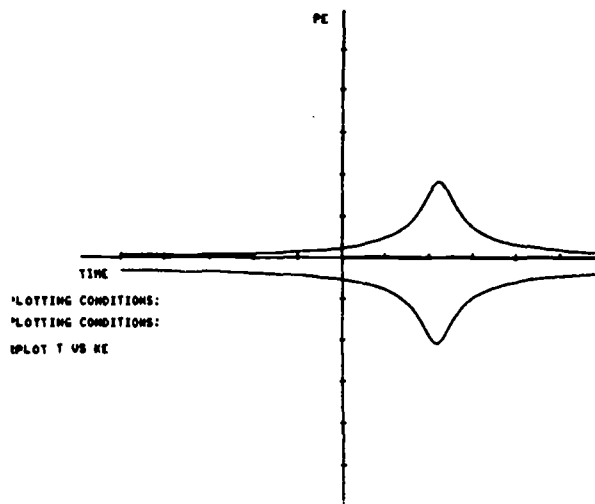
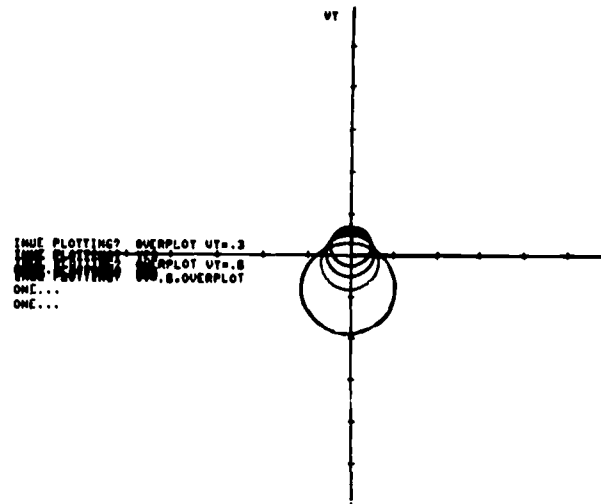


Figure 3

Figure 1: Inverse square force, varying initial velocity

Figure 2: Inverse square force, velocity space, varying initial velocity

Figure 3: Inverse square force, kinetic and potential energy as functions of time

oriented toward easy manipulation of graphic material are also array-oriented languages. However, these specialized languages are available in only very few places.

Of the common general purpose languages we might use with physics students, the one presenting the best array capabilities and therefore the one most suitable for graphics is APL. APL has other advantages as an introductory language for students, making it the language of choice if all current languages were available in a given location[10].

Since APL can use array arguments to functions, several natural ways for graphics are available. Some experimentation with running systems will be useful for determining which of these is both most natural for the experienced APL user and easiest to manipulate for the beginning user. If A and B are arrays of the same length containing the data the following APL command, for example, might generate the graph:

DRAW A VS B

Just as in dialog graphics, we need windowing and scaling, so additional functions are also necessary. And 3-D graphing should also be allowed[11].

We hope to have a running APL graphic system soon, so we can gather experience with students. In generating teaching material we believe it essential to interact at all stages with students and to adapt the form and structure in ways that are amenable to them, rather than forcing them to match the software.

Conclusion

We have reported on graphic development and plans within the Physics Computer Development Project. Some of the ways outlined are dependent on the physics teaching material, so other areas may find other modes more natural. It may turn out that graphics are not more effective than cheaper methods of computer output in some areas. We believe that in physics the case for graphics is already strong and we believe that the potentialities for the future are great. We encourage others to experiment both within physics and in other areas to learn the capabilities of graphic teaching materials.

REFERENCES

1. Bork, A., The Computer in a Responsive Learning Environment-Let a Thousand Flowers Bloom.
2. Bork, A. and Ballard, R., The Physics Computer Development Project.
3. Bork, A. and Mosmann, C., Teaching Conversations with the XDS Sigma 7--System Description.
Mosmann, C. and Bork, A., Teaching Conversations with the XDS Sigma 7--System Users Manual.
Warner, E. and Bork, A., Teaching Conversations with the XDS Sigma 7--System Maintenance Manual.
4. Bork, A., Warner, E., and Collins, J., Teaching Conversation with the XDS Sigma 7--Graphic Dialog Facilities.
5. Bork, A., Terminals for Education.
6. Bork, A., Inexpensive Timeshared Graphics with the Sigma 7.
7. Bork, A., Luehrmann, A. and Robson, J., Introductory Computer-Based Mechanics.
8. Taylor, E. and Wheeler, Jr., Spacetime Physics.
9. Bunderson, C., Instructional Software Engineering.
10. Bork A., Science Teaching and Computer Languages.
11. Bork, A., Graphics in APL.

STATISTICAL PHYSICS COMPUTER APPLICATIONS

Harold Weinstock
Illinois Institute of Technology
Chicago, Illinois 60616

Introduction

Statistical mechanics is a subject which is of fundamental importance in a great variety of areas of physics specialization. Yet, it embodies a number of abstract concepts which prove difficult for students to master, and because of calculational complexities or time limitations, only a select number of sometimes unrealistic and trivial problems can be assigned (for examinations or even homework). The computer can be most helpful in broadening the range of solvable and meaningful problems.

While this last statement may be made with validity about most areas of physical science, it is particularly appropriate for statistical mechanics. This subject involves calculation of macroscopic thermodynamic parameters by taking suitable averages over a large collection of microscopic entities which comprise a system. Inasmuch as the fundamental postulate of statistical mechanics states that a system in equilibrium may, with equal probability, be in any of the physical states accessible to it, a number of computer simulations can be made which will provide accurate representations of actual physical systems. The use of random number generation figures prominently in such simulation. The "number-crunching" capability of the computer is, of course utilized both in simulations and in the evaluation of complex mathematical relations.

In this paper I wish to report on two homework assignments I have made to students in senior and graduate level courses. These are assignments which were made after I had motivated class discussions on hopefully meaningful problems not normally handled. While each of these assignments could have been completed without the aid of a computer, in one case the computation would have been prohibitively time consuming, and in the other case, most students did not have the mathematical sophistication and intuition to solve the problem. However, both assignments involved elementary programming skill and required a relatively limited amount of time to find solutions. All (15) students involved were able to complete these assignments with some measure of success. These students ranged from junior class physics majors to graduate students, all of whom had previously been exposed to at least one introductory programming course. The programs and output to be presented below represent upgraded versions of some of their efforts. For students in an introductory physics course, this material could be used in subroutine or "canned" form. An interesting fact about both these problems is that the students found the results unexpected in almost all cases considered, while even I was startled by some of the results.

One Dimensional Random Walk with Unequal Probabilities

As an introduction to statistical fluctuations and distributions, the one dimensional random walk problem is generally presented. It is observed that one can describe such a chain of events by a binomial distribution of the form:

$$P_n(N) = \frac{N!}{(N-n)!n!} p^n q^{(N-n)}$$

where $P_n(N)$ is the probability of moving n positive, e.g., right, steps in a total number of trials (or steps) N ; p is the probability of moving in the positive, e.g., right, direction for a given trial; q is the probability of moving in the negative, e.g., left, direction for a given trial; and $p + q = 1$ for all possible situations for which the distribution applies.

The usual example given involves a drunk starting out at some origin and (unrealistically) moving randomly either forward or backward with equal probability ($p = q = 1/2$) in taking steps of equal length along a straight line path. For a given total number of steps N , it is easy to use the above equation to calculate each $P_n(N)$ and arrive at the symmetric discrete binomial distribution in which the most probable position is at the origin, assuming N is an even number. This result seems to appeal to everyone's intuition and is expected.

Rarely, however, are the consequences of a non-symmetric, $p \neq q$, binomial distribution investigated even though, as it turns out, its consequences are unexpected and have significance with regard to the motion of wave packets.

To remedy this situation, I challenged my class to tell me where I should place the drunk in a linear sequence of 31 squares, number 0 to 30, such that he would arrive safely at either the 0 or 30 square with equal probability when $p = 2/3$ and $q = 1/3$. As expected, the class unanimously agreed that square 10 should be the correct starting position, i.e., that position which divides the total number of squares into two segments whose ratio is that of $p/1$. Rather than take advantage of the student's naiveté in playing a related game based on my having the $p = 2/3$ probability of a successful trial, I presented them with the following problem:

0	1	2	-----	X	-----	28	29	30
---	---	---	-------	---	-------	----	----	----

"Assuming a random distribution of independent events with $p = 1/3$, $q = 2/3$; a field of play with 31 positions labeled consecutively from 0 to 30; a move of one unit to the left with a 'p' result and vice versa for a 'q' result; a win for you when the zero position is reached; and a win for me when the 30 position is reached; what number should you designate as the starting position to make the game fair? Solve this problem using standard mathematical techniques and do a simulation of it on the computer 'playing' at least 10 'games' at the position you think (or have found) to be the fairest."

All of the students in the class were able to do the programming necessary to carry out the assignment although the degree of sophistication varied. Yet, only two of them were able to solve the problem by standard techniques using a recursive relationship[1]. Even then the solution was not in closed form as it required the entry of that starting position which would produce an equality or near equality. Thus, the computer simulation was useful in providing the correct region for the starting position, if not in fact the correct answer by virtue of a sufficiently large number of simulations.

To dispel any lingering curiosity about the outcome, let me point out that the fairest starting value is found (both mathematically and by simulation) to be position 1. Starting at that point, the probability of reaching position 30 is just a bit greater than $1/2$. For example, in one hundred trials a student found 52 trials which ended at 30.

Figure 1 illustrates a convenient form of output for this problem. Shown are the random numbers generated by the computer and the resulting position (of the drunk) for a given starting position. The students soon realize that with a starting position of 1, their chances of success in the game, i.e., reaching position 0, diminish rapidly if the initial step is in the positive direction. To keep the total output down to a reasonable size, only the first ten games in a sequence of 100 for a given starting position are plotted. Then a summary of wins for each side is given.

Although this exercise was introduced within the context of a statistical mechanics course, I believe it is just as significant as an illustration of the statistical foundation of quantum mechanics. The non-symmetrical binomial distribution can be considered as characteristic of the behavior of a particle acting under the influence of a uniform force field while subject to random collisions. For example, this situation would apply to a charged particle in a gas acted upon by an electric field. It can be said that while the most probable position of a particle moves in the direction dictated by the applied force field, its wave function spreads out in space. Thus, there is always a non-zero probability that the particle will have moved in the opposite direction. This probability diminishes rapidly once there has already been some "drift" in the opposite direction.

The programming skills required to perform the above described simulation is elementary and well within the capability of an upperclass physics major or graduate student. A flow chart of the program used by the author is presented in Appendix A.

Heat Capacity for Multi-Level Energy Systems

A fairly standard problem is one which involves calculation of the heat capacity of a system of N classical particles with atomic spin $S = 1/2$. Quantum mechanical considerations, which must be applied, dictate that there are only two energy states available to each particle (corresponding to oppositely directed spin orientations). Calculation of the total energy of the spin system, and subsequently its heat capacity is straightforward and simple. The average total energy of the system (\bar{E}) is determined using the general expression

$$\bar{E} = N \frac{\sum_i \epsilon_i g_i e^{-\beta \epsilon_i}}{\sum_i g_i e^{-\beta \epsilon_i}}, \text{ where } \epsilon_i = \text{the energy of the } i^{\text{th}} \text{ level,}$$

$$g_i = \text{the degeneracy factor of the } i^{\text{th}} \text{ level}$$

and $\beta = 1/kT$.

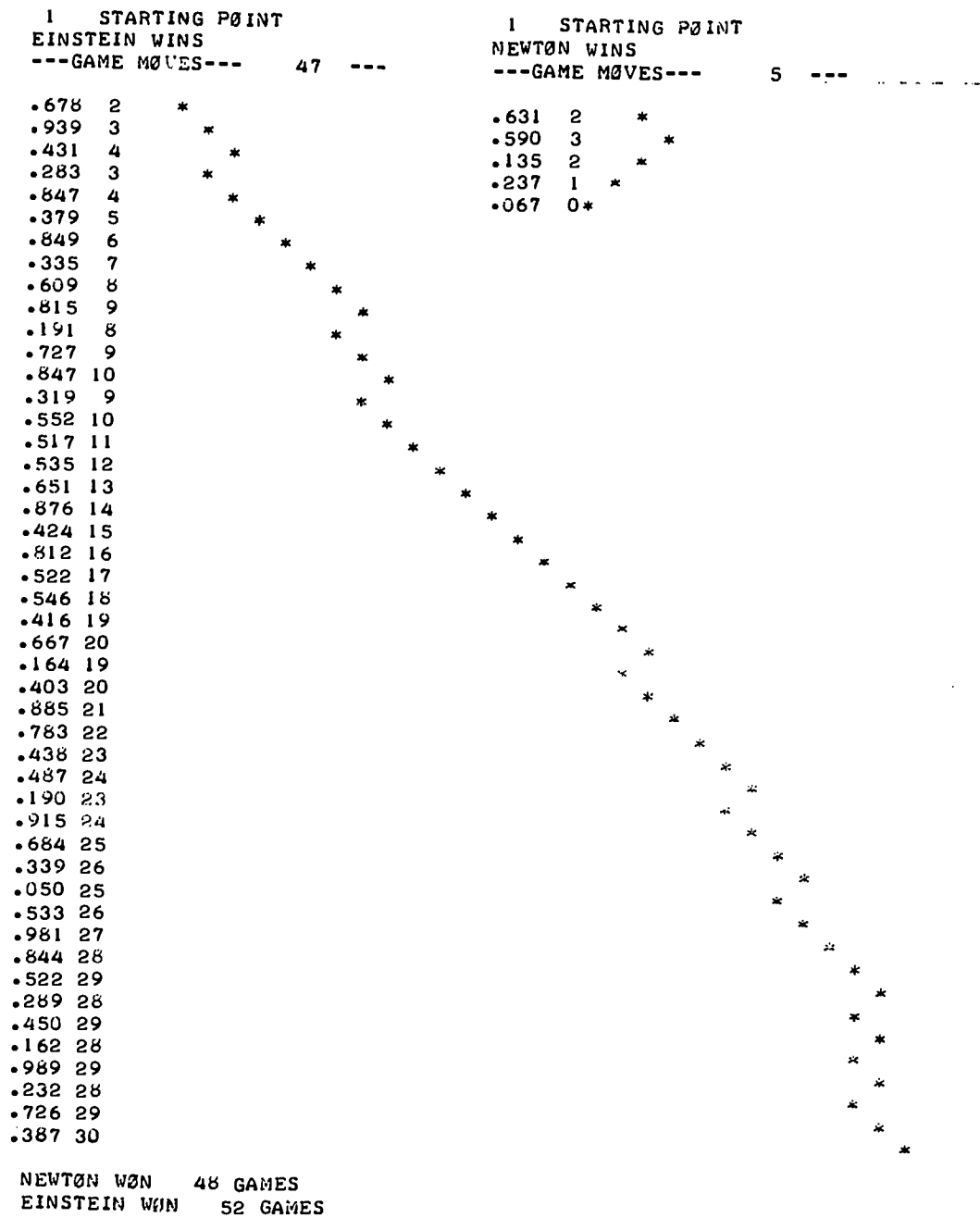


Fig. 1 - Selected sample output for random walk problem with $p = 2/3$, $q = 1/3$.

418



This expression is valid for any system represented by a discrete number of levels. The heat capacity at constant volume (C_V) then is obtained using the fundamental relation

$$C_V = \left(\frac{\partial E}{\partial T} \right)_V$$

yielding,

$$C_V = \frac{N}{kT^2} \frac{\sum_i \epsilon_i^2 g_i e^{-\beta \epsilon_i} \sum_i g_i e^{-\beta \epsilon_i} - (\sum_i \epsilon_i g_i e^{-\beta \epsilon_i})^2}{(\sum_i g_i e^{-\beta \epsilon_i})^2}$$

For a two level system, one rarely bothers using such rigorous formalism to evaluate C_V . Since the value for C_V depends only on the difference in energy between levels, one can assign a value of 0 to the lower level and a value of ϵ to the upper one. Equation 1 then becomes[2]

$$\bar{E} = N \frac{\epsilon e^{-\beta \epsilon}}{1 + e^{-\beta \epsilon}}$$

Differentiating this directly with respect to T gives

$$C_V = \frac{N \epsilon^2}{kT^2} \frac{e^{-\beta \epsilon}}{(1 + e^{-\beta \epsilon})^2}$$

Plotting this expression as a function of temperature, shows that at low temperature ($kT/\epsilon \ll 1$) the heat capacity rises exponentially. In the vicinity of $kT/\epsilon = 1/2$ it peaks - the so-called Schottky anomaly peak - and then falls off as $1/T^2$ for higher temperatures ($kT/\epsilon > 1$).

While there are numerous physical systems for which the above discussion applies, there are obviously many more which are composed of more than two energy levels. As a homework problem I proposed that my students investigate a three level system (appropriate to many solid state lasers) for which the level energies are 0, ϵ , and 3ϵ , and further suggested that they present a computer tabulation and graph of the resultant C_V vs. T . To almost everyone's surprise, it was found that there is still only one Schottky peak, again in the vicinity of $kT/\epsilon = 1/2$, although there is a quantitative difference between results for a two level system and the particular three level system chosen here.

The problem of a three level system can be posed in yet another way, one which has considerable relevance to interpretation of actual physical measurements. Suppose an electron spin resonance experiment shows two resonances, one at twice the frequency (or energy) of the other. Eliminating the possibility that the higher frequency resonance occurs for a transition from the lowest to the highest level, there are still two possible level configurations, either 0, ϵ , and 3ϵ or 0, 2ϵ , and 3ϵ . Figure 3 shows the graphical output of C_V vs. T for these two distinct level configurations. Also shown in Figure 2 are results for 0- ϵ - 5ϵ and 0- 4ϵ - 5ϵ configurations. Note that for the 0- ϵ - 5ϵ configuration some evidence of a second Schottky peak is observed.

The computer program used to produce Figure 2 and subsequent figures is basically simple and inexpensive. A flowchart of the program is presented in Appendix B. Its major function is to evaluate Equation 3 for the given level configurations. Output is presented in both tabular and graphical form with up to four level schemes per table and graph. Input includes the level configurations to be evaluated, the temperature interval and whether the levels are equally or unequally spaced - the reason for this last piece of information will become evident from a discussion which follows. If level spacing is not equal, input is limited to 10 levels. If the converse is true, up to 50 levels may be considered. The ordinate axis is plotted in terms of the dimensionless parameter C_V/Nk and is scaled from 0 to 1. The abscissa is in terms of the dimensionless parameter kT/ϵ with the scaling specified by the input.

Another interesting use of this program involves computation of C_V vs. T for a series of configurations with equal spacing, but with an increasing number of total levels. Figure 3 shows the resulting curves for 2, 5, 10 and 30 equally spaced levels. As one goes from the familiar 2 level Schottky anomaly to the successively higher leveled curves, it is seen that the peak occurs at higher temperatures and has a higher C_V/Nk value, and it also is seen that as the

X .000 1.000 5.000
 O .000 1.000 3.000
 + .000 4.000 5.000
 * .000 2.000 3.000
 PLOT OF C/MK VS. KT/EPSILON. ORDINATE FROM 0.0 TO 1.0.

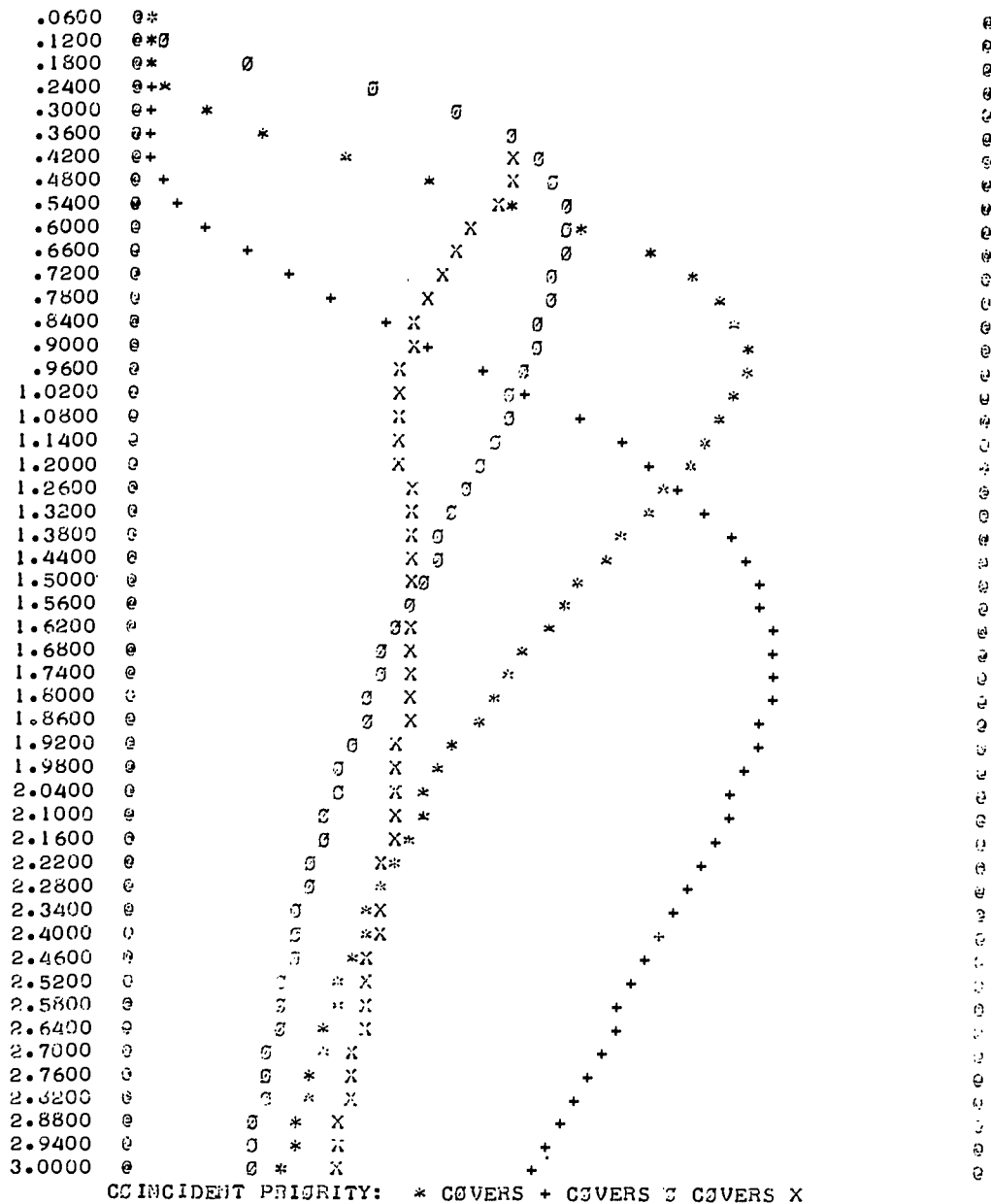


Fig. 2 - Heat capacity vs. temperature for four unequally spaced three level systems:
 (0,ε,3ε); (0,2ε,3ε); (0,ε,5ε); and (0,4ε,5ε).

UNIFORMLY SPACED LEVELS OF: 2 5 10 30
 DENOTED BY: X 0 + *
 PLOT OF C/NK VS. KT/EPSILON. ORDINATE FROM 0.0 TO 1.0.

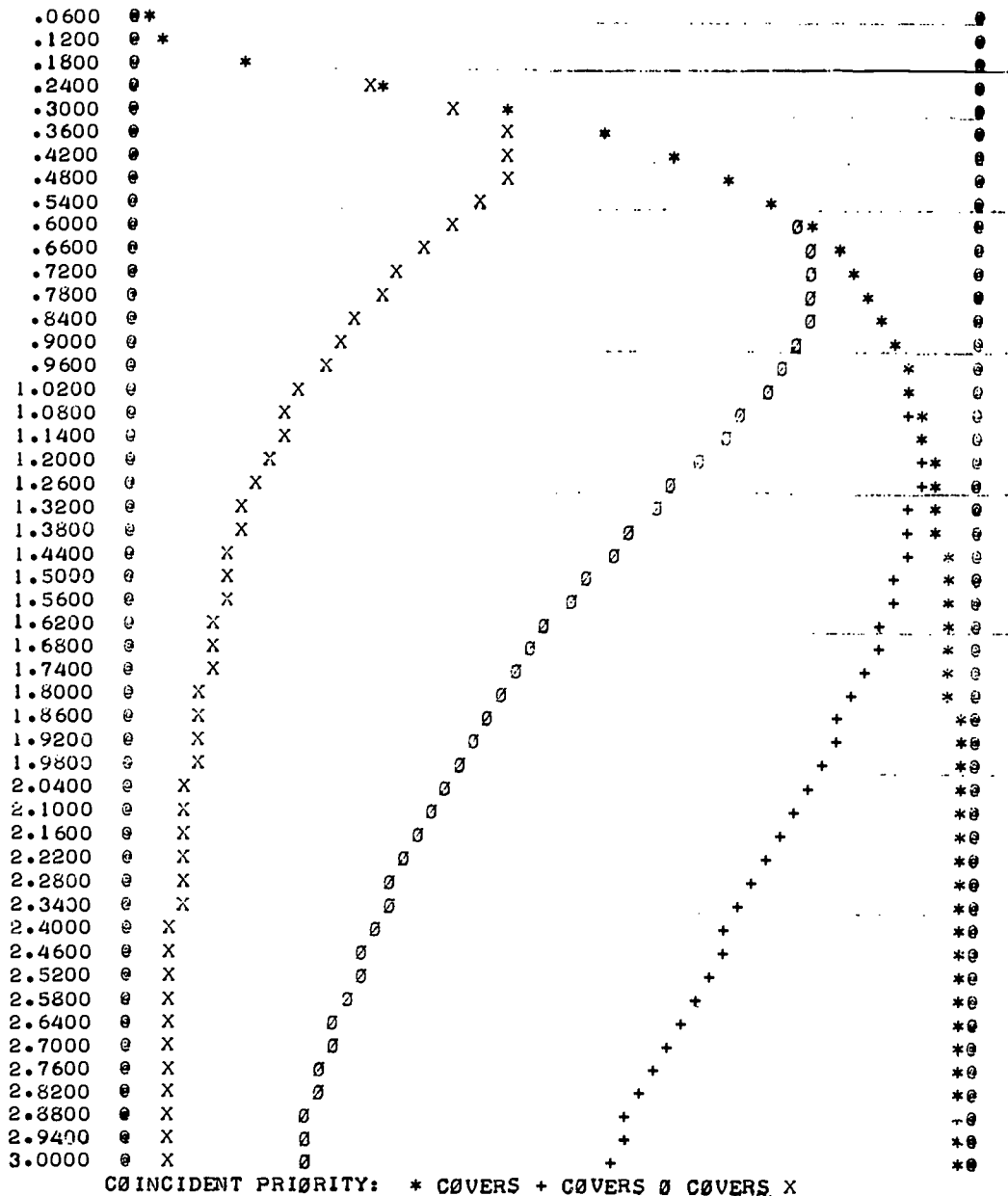
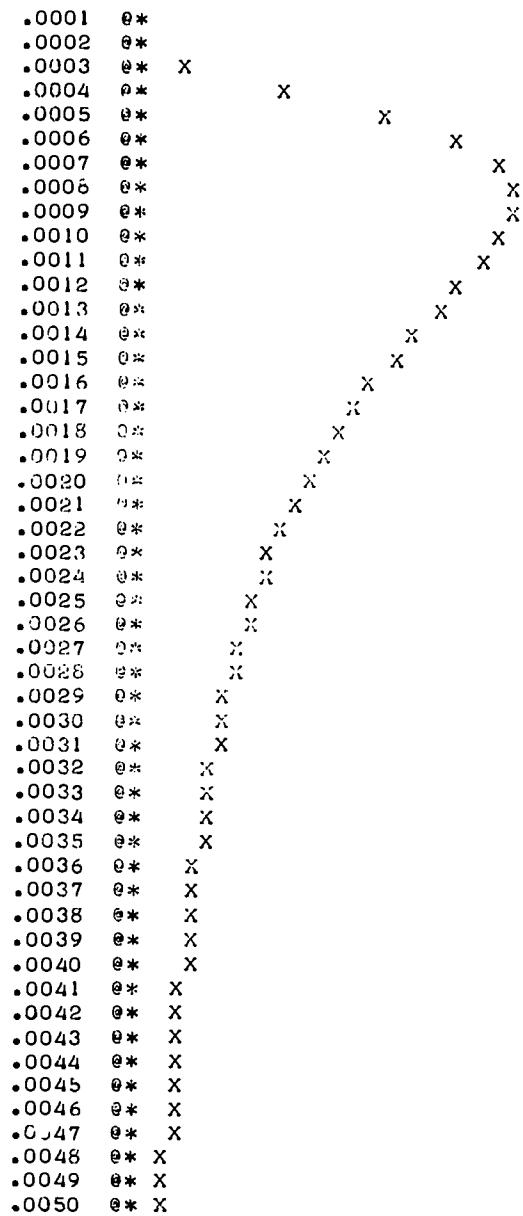


Fig. 3 - Heat capacity vs. temperature for four equally spaced level systems of 2, 5, 10 and 30 levels.

X .000 .002 1.000 1.002
 0 .001 1.001
 PLOT OF C/NK VS. KT/EPSILON. ORDINATE FROM 0.0 TO 1.0.



COINCIDENT PRIORITY: * COVERS + COVERS 0 COVERS X

Fig. 4 - Heat capacity vs. temperature for a simple two level system and a similar system showing hyperfine splitting for doubly degenerate atomic levels. Note expanded temperature scale.

number of levels becomes quite large, C_V/Nk approaches 1. Such a result is nothing more than an approximate numerical solution for the heat capacity (for one dimension) of an Einstein solid, i.e., a solid in which it is assumed that each atom vibrates as a harmonic oscillator with a fixed frequency. There are an infinity of energy levels given by $E_n = (n + 1/2)h\nu$, where n may be any positive integer. The ground state energy of $1/2 h\nu$ (or $1/2 \epsilon$ in the units used here) is a consequence of quantum mechanics which in no way affects the calculation for heat capacity. The usual method for calculating the Einstein heat capacity[3] is to utilize a mathematical identity for the sum of an infinite series with geometric progression. Utilizing a computer generated numerical solution, a student can take a more active role in arriving at the desired result and perhaps gains a stronger physical feeling for its significance.

One final application of the program based on Equation 3 is illustrated in Figure 4 for a two level system with doubly degenerate levels split by a hyperfine interaction. To simulate a physically meaningful situation, the resulting four levels due to hyperfine interaction are taken as 0, 0.002ϵ , ϵ , and 1.002ϵ . Also evaluated for comparison is the unsplit (0, ϵ) two level system. One sees that the four level split system exhibits a nuclear Schottky anomaly identical in form to that observed for a purely atomic situation. Both the split and unsplit systems show the same behavior in the temperature region of $kBT/\epsilon \sim 1$. Students must be careful in choosing the scale(s) to be used in handling this problem, as they could easily miss seeing the nuclear Schottky anomaly and falsely conclude that hyperfine splitting does not modify the heat capacity.

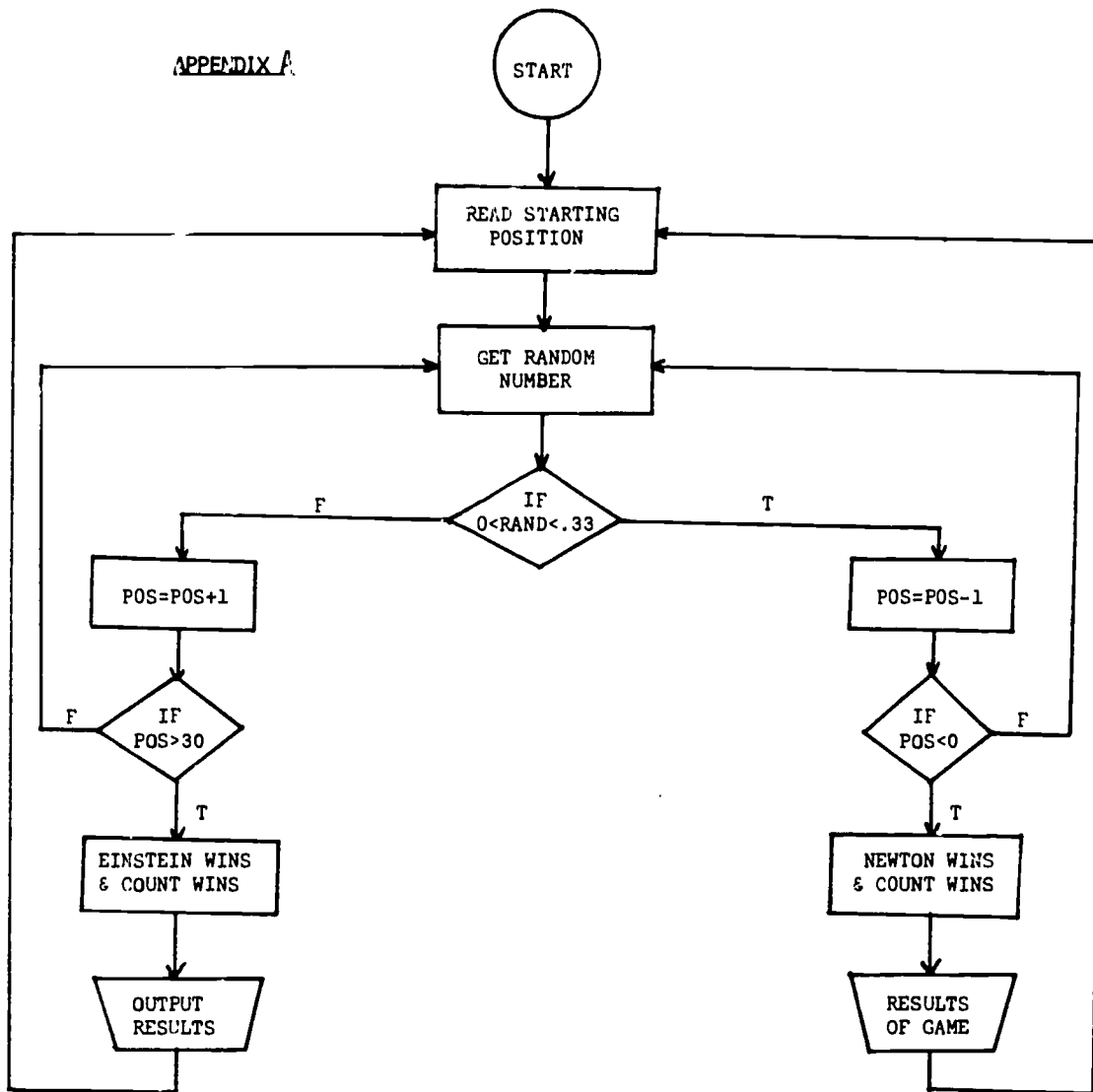
REFERENCES

1. See W. Feller, Introduction to Probability Theory and Its Applications (Wiley, New York, 1957) Chapter 14, for a discussion of this problem.
2. For the purposes of this paper, it will be assumed that all levels have the same degeneracy, and hence that the degeneracy factor can be ignored.
3. See F. Reif, Fundamentals of Statistical and Thermal Physics (McGraw-Hill, New York, 1965) p. 254.

1
1
3
1
7

Flow Chart for Random Walk Problem

APPENDIX A



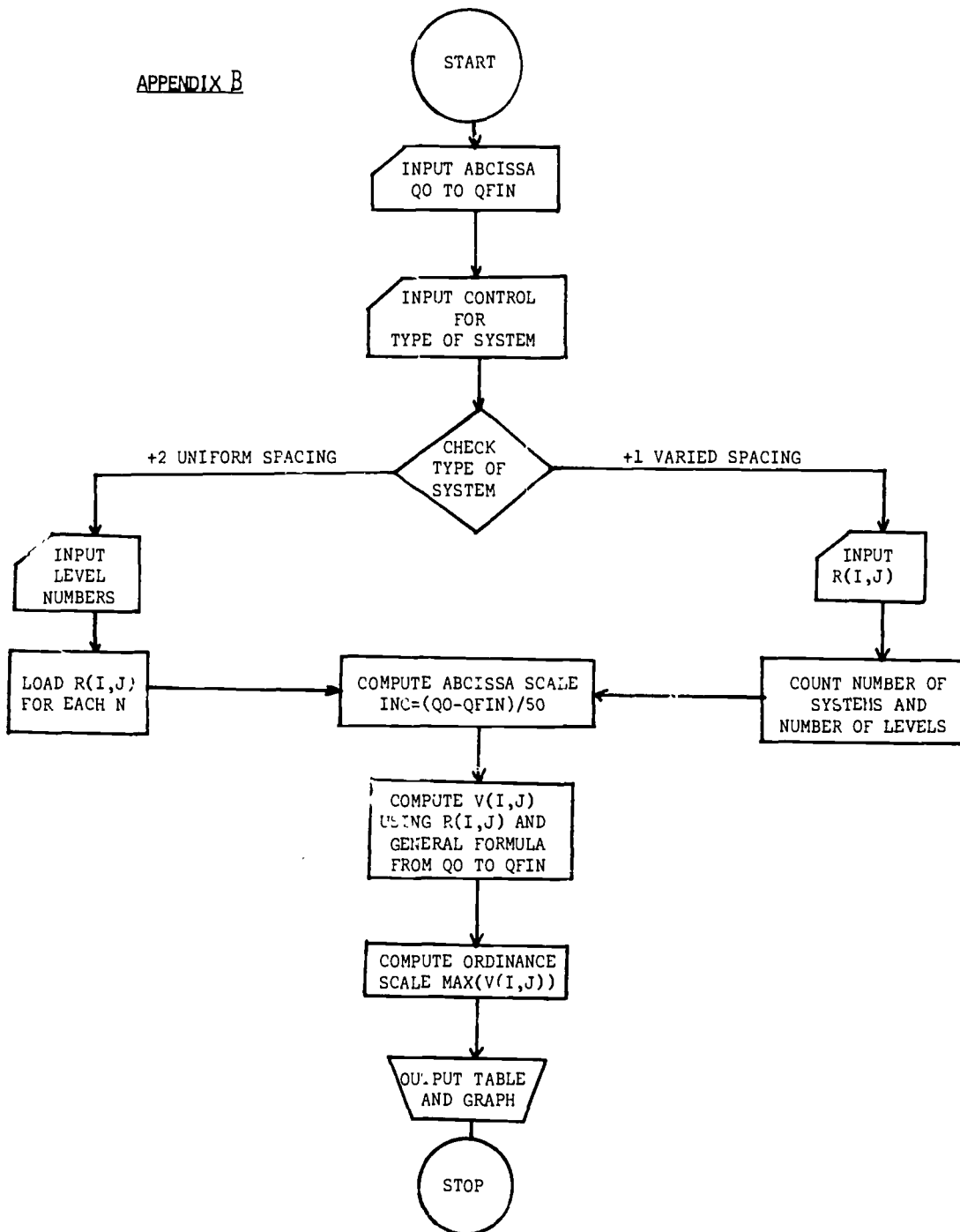
$P = \frac{1}{3}$ results in a -1 step

$Q = \frac{2}{3}$ results in a +1 step

Absorbtion limits of 0 and 30

Flow Chart for Schottky Anomaly Heat Capacity Program

APPENDIX B



NOTE: Always Maximum of 4 systems with 50 levels/system₁ and 10 levels/system₂

AN INTERACTIVE COMPUTER TEACHING DIALOG
FOR SOLVING A SYSTEM OF COUPLED OSCILLATORS

Charles P. Munch
University of California, Irvine 92664
Telephone: (714) 833-6911

ABSTRACT

In this paper the theory and development of an interactive teaching program is documented. The efforts extend over a period of one and a half years and involve usage by a class of undergraduate science and engineering students. Revision on the basis of student feedback is emphasized. At the time of the 1972 Atlanta conference, the program will have had a second usage by the computer based mechanics course.

Introduction

Computer usage in the physics curriculum at Irvine has been well documented in several places[1,2,3,4]. The Physics Computer Development Project is the primary purveyor of these applications. Interactive teaching programs or dialogs are a major output of the project. The purpose of this paper is to report on one of the larger and more involved dialogs that we have produced. I discuss some of the relevant technical and pedagogical aspects in the production of such materials.

The program, named COUPOSC, is intended for use with a computer based mechanics course aimed at science and engineering freshmen. This course was the subject of Alfred Bork's paper at the previous conference in Dartmouth.

Choosing A Topic

One aspect in the production of computer teaching materials that warrants careful consideration is the choice of topic. Interactive teaching programs are involved and costly affairs. To maximize effectiveness, we must consider where the computer will offer the greatest advantage.

The decision to write a program on coupled oscillators was based on several matters. Areas of the course where students had previously encountered difficulty were examined. The combination of new mathematical concepts and new conceptual requirements involved in coupled systems made this a weak area for students. Classroom and text treatments seem insufficient for many students in developing the notions required for solving problems. In addition, coupled systems were a transition from mechanics to the study of waves, making the concepts involved crucial to the students' progress.

It was felt that the major inadequacy of classroom or text treatments was that the student was not involved in the derivations. As a passive experience, the student would have more difficulty in developing the notions than if he interacted directly. Often it is found that even if a student is able to follow the steps in a derivation, he is unable to reproduce it or, more importantly, generalize on his own.

With a carefully written dialog, students would be allowed to be instrumental in the solution of the problem, discovering for themselves what a normal mode was and then being told its name. With the program offering guidance and remedial help most students could have the satisfaction and educational advantage of having solved the problem themselves, learning the concepts as they proceed.

A student having learned this way is at an advantage in solving related problems. In attempting to generalize from material obtained in a passive situation, a student may find there are many points where his understanding is insufficient. The reasons behind choices in a derivation may not be explicitly stated, say in writing the equations of motion or in choosing the normal coordinates. For a student who learns by a dialog, his misunderstandings are exposed and can hopefully be identified by the nature of the responses. In any event, it is felt that the student should gain a sounder grasp of the material when his exposure has developed as a function of his thinking.

The Pedagogy of the Interactive Derivation

The idea of an interactive proof originated with a dialog for the derivation of the conservation of energy, documented elsewhere[4]. This author was active in the development of that program so much of the style and pedagogy is reflected in the program presently being discussed.

To give a better idea of how the program is structured, a brief flowchart is given in Figure One. To ensure that the student has the necessary background, a brief review of spring forces and simple harmonic motion is the starting point. The simplest case for the system is taken. We consider two bodies of equal mass riding in a linear array of spring-body-spring-body-spring. In the program the notions of degrees of freedom, normal modes, normal coordinates, characteristic frequencies and configuration of the system in a normal mode are developed.

The objective stated to the student is to find a general description of how the masses move. The author's goals can be stated as short and long term. As a short term objective, we wish the student to grasp various concepts. These notions along with some factual information comprise the material the student needs at a particular point in the course. What is a more long range benefit and what can ideally be better realized by a dialog is a skill in handling new and difficult problems.

If the program is going to realize the objective of allowing a truly interactive derivation, care must be taken in designing what the student will be shown and what he will be asked. In the program being discussed, the student is allowed to make all the important decisions relevant to the course of the program and to give all the important results. The responses sought from the student range from asking for very specific information such as a formula, to quite general and open ended queries such as "how do we proceed from here?" The program is designed so that the student is never presented with long, uninterrupted texts unless it is requested supplementary information. Frequent student interaction is necessary throughout the program.

One of the easiest ways students become discouraged is when a program fails to recognize their correct responses. A major effort was devoted to matching sequences that allowed as wide a variety of correct responses as feasible. Of course, an author cannot anticipate all forms of correct responses so revision on the basis of student feedback is essential. Revision will be dealt with in depth later. Matching sequences were structured to allow the student latitude in the format of the input. Restrictive rules on input format can only be a hinderance in the educational experience.

At important points in the program where a correct response is not found, extensive remedial help is available. A student is never told that he is wrong. Depending on the nature of the question, a student is branched to a number of possible sequences.

In some places, after an unsuccessful match the student is given increasingly specific hints and is eventually told the answer or choices he has in how to proceed. In other spots, where the particular weakness can be identified, the student is taken through a remedial sequence, usually being asked simpler questions relating to the difficulty encountered. Much effort is devoted to the recognition of incorrect responses since this allows the author to respond directly to the particular difficulty the student is having. This recognition sometimes takes the form of identification of missing or incorrect elements in the response.

Suggestions and remedial help are given to a student only when responses indicate a need. Thus, for example, the student having difficulty in the sequence where the equations of motion are entered can be exposed to several pages of material lasting thirty minutes. A student who can immediately write down the answer may see six lines and take one minute.

Our system has extensive capabilities for storing and sorting student inputs and these are used to their full extent in this program. Due to its size, it is impractical to save all student input from the program. A save command is used at those inputs where the response would be indicative of the student's progress or where necessary revision is anticipated. According to the placement of the save command, a student's input is either stored unconditionally or only if a correct response is not found. Fifty inputs use the save command in the program.

Development

The production of the first version of the program extended over several months. Its development can be traced through various stages. The initial stage has already been mentioned, that being the choice of topic. Following this, research was done in several mechanics texts to reveal the varying methods and aspects of the solution. A general plan was formulated and then sketched in flowchart form. Next, the major portions of the text were written on paper. This was incorporated in the production.

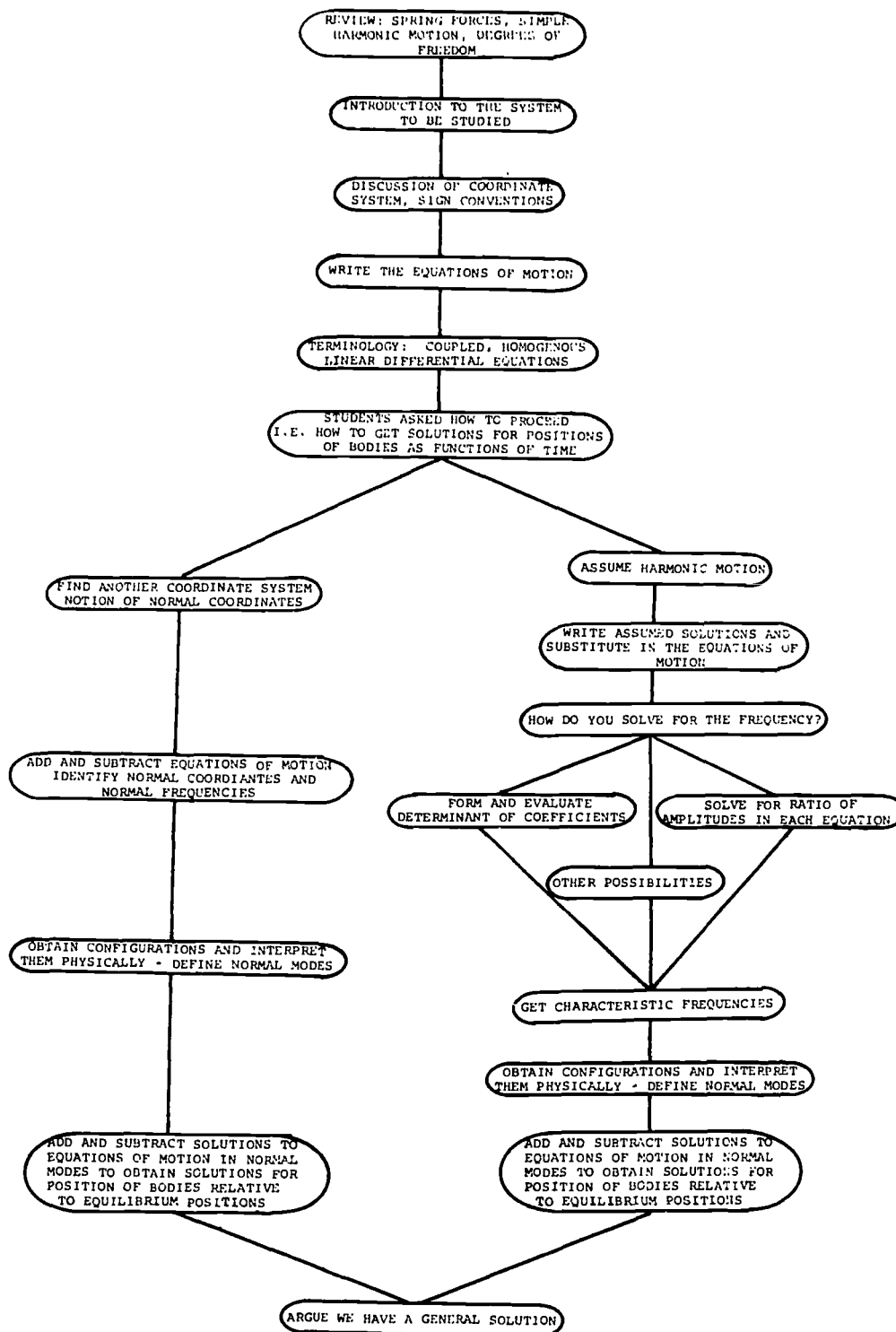


FIGURE ONE

of a complete flowchart where all branches and text were diagramed. The full flowchart, twenty-five pages of large paper, took about two weeks.

Discussion of the material with colleagues was used at all stages and on the basis of comments on the full flowchart a revised flowchart was made. A more explicit indication of the directives was included. A secretary used this material to type in the code for the program which was stored on disk. This is the standard procedure for the project and is described in detail elsewhere [5]. It took the secretary about a week to enter the bulk of the material, the file containing around 2,100 lines. The author then began debugging and filling in portions of the program. Because of its size, the file was broken into several smaller files, the advantages being that the smaller files could be corrected and reassembled individually and that with an overlay structure the entire program need not occupy core at one time. A new file was created for linking the various files.

It took approximately one working month to obtain the first running version of COUPOSC. This implies that all syntax and programming errors had been eliminated but says nothing of grammatical or logical errors. The program was executed repeatedly by the author, members of the project and visitors. This uncovered most obvious errors and omissions. The program was constantly corrected for these over the initial testing period of about two months.

At the end of this period the program was still far from finished. The usage up to this point had not included a substantial number of students to whom the material was fresh, but instead has involved faculty or students already knowledgeable on the subject. Usage by virgin students is the only way the program can be evaluated and revised effectively. Revision based on student responses requires scrutiny of the feedback and is the only way that truly responsive and viable dialogs can be developed.

Student Usage and Feedback-Based Revisions

The first class exposure was in the spring 1971. The program was optional, one of several alternative ways of learning the material. The response files indicated about forty users in that time period. The material stored by the save commands was voluminous.

Careful analysis of this data is necessary to uncover points in the program where there are pedagogical weaknesses, questions which are unclear to the student and points where more assistance is needed. It must be realized that the author's own way of thinking is very much reflected in the material he produces. Aside from the difficulty of spotting one's own mistakes, the weaknesses or obscurity of a particular sequence in a dialog may be invisible to the author.

The data available concerning student performance is of three types. Free form student comment provides the most direct information on where the program has difficulties. This comment can come either at the end of the program, where comments and suggestions are solicited or in questionnaires handed out in class. Comments are delivered in person also. Secondly, student inputs (or lack of) are an indication of their performance. A third useful datum is provided by counters. These are stored internally and allow us to see how many times a student repeated a question.

If response information indicates that a large number of students have difficulty with a certain input, analysis must be made to uncover how the program is failing. Possible explanations are: (1) a correct response is not being recognized, (2) the question is unclear to the student and (3) there has been inadequate or insufficient preparation for the question. Most revisions were based on the first type of difficulty. However, it was felt that the most important revisions dealt with the latter two categories. All categories will be discussed individually with examples.

A majority of the revisions done on the basis of student feedback were the recognition of new forms of the correct response. The revision sometimes involved simply the addition of new strings to be recognized. In places where the matching structure is inadequate to handle a variety of responses, the sequence may have to be entirely rewritten, making it more complex and versatile. For free-response questions, examinations of student input for unanticipated responses is essential since there will invariably be a myriad of ways of stating the answer which cannot be anticipated. Likely or indicative forms of incorrect answers should be sought and responded to.

An example in COUPOSC of an input where the addition of new words to be matched was sufficient is a point where the student is asked how the masses move relative to each other in the lower frequency normal mode. Originally, only five words were recognized. During the initial usage, many other possibilities were discovered and now there are over twice as many possible correct responses.

429

A more involved revision came at a crucial point in the program. The student was asked how he would like to solve the set of coupled differential equations of motion. In the first version, several words and combinations of words were recognized as suggesting possible methods of solution. The coding took about fifty lines. On the basis of student data a more complex and comprehensive sequence was devised. The coding related to this input now numbers around 200 lines.

Questions of a general or open-ended nature tend to bewilder the student who is not following the material very successfully. The student sees the question as nebulous and has no idea of how to respond. Inputs at these points often show things like "help" or "I'm lost". It is difficult to anticipate such points and so analysis of student data is necessary to uncover such spots.

We consider a particular input in COUPOSC as an example. After disclosing the objective of finding out how the system moves, the student is asked how to proceed. That is, what shall we do to find out how the system moves. It was found that many students did not know how to respond to this question. Though students might realize that the equations of motion would be needed somewhere in the solution it was not clear to them that this was the starting point. Thus, it was felt that a more specific and suggestive wording of the question was needed. This would allow more students to guess the answer without having to be told. In the revised edition the student was asked to complete the sentence, "In mechanics, we usually begin by writing..." Subsequent use of the program indicated a substantial improvement in student response.

Inadequate preparation leading up to a question is also a cause of poor student response. The nature of the mistakes in the input can be indicative of what the student is missing, and material to clear up the uncertainty can then be included. Whether the material is included as a remedial sequence, or whether it is preparatory material which all students are exposed to depends on the nature of the material and the number of students showing a need. We try to strike a balance between exposing students to material which they might already know and preventing some students from even attempting a response because of inadequate preparation.

As an example, it was found that when students were asked to enter the equations of motion, uncertainties about the number of forces acting on a single body and the dependence on relaxed length of the spring were exposed. A brief section asking about these issues was thus included just before the question. Hopefully, any confusion about the matter is now exposed and corrected prior to writing the equations of motion.

Conclusion

What I have attempted to convey in this paper is that writing a dialog is a sizeable task which requires careful consideration and conscientious revision in order to produce materials which are pedagogically sound. It is felt that the dialog method of learning offers promising hope for education, but caution must be exerted. Efforts in this direction may represent no improvement over the present situation or may even represent poorer methods of teaching if the materials do not reflect the author's diligence at making them responsive and tailored to the student.

REFERENCES

1. Bork, Alfred H., "Computer-Based Mechanics," Proceedings of Conference on Computers in Undergraduate Science Education. Published by Commission of College Physics, College Park, Maryland, 1971.
2. Bork, Alfred H., "The Computer in a Responsive Learning Environment--Let a Thousand Flowers Bloom," Proceedings of Conference on Computers in the Undergraduate Curricula, Dartmouth College, Hanover, N.H., June, 1971.
3. Monroe, Mark, "Physics Computer Development Project-Computer Assistance in Student Problem Assignments," Proceedings of Conference on Computers in the Undergraduate Curricula, Dartmouth College, Hanover, N.H., June 1971.
4. Bork, A., and Sherman, N., "A Computer-Based Dialog for Deriving Energy Conservation for Motion in One Dimension," American Journal of Physics.
5. Bork, A., Ballard, R., "PHYSICS COMPUTER DEVELOPMENT PROJECT," University of California, Irvine.

PROJECT CAPLIN: COMPUTER AIDED PHYSICS LABORATORY
INSTRUCTION FOR NON-SCIENCE MAJORS

W. E. Bron
Indiana University
Bloomington, Indiana 47401
Telephone: (812) 337-1304

Introduction

We have recently completed a set of computer aided laboratories for an introductory physics course for non-science majors. Laboratories in the areas of kinematics, mechanics and wave motion have been developed and tested on large numbers of students. Laboratories in the areas of electricity, magnetism, thermodynamics and selected advanced topics are to follow.

The laboratories contain what we believe to be unique features in conception and execution. The following primary tenets serve to describe the approach taken in the development of the computer material. (1) Computer material is used in conjunction with traditional laboratory experiments used to illustrate physical concepts and laboratory practices. In this environment the computer is meant to enhance the student's laboratory experience; not to supplant it. (2) The laboratory is aimed primarily at non-science majors (predominantly pre-medical students in our case). Consequently, the computer material must take into account the student's disinclination toward (and often disinterest in) mathematical and physical manipulations. (3) The interaction of the student with the computer should closely resemble the student's most probable mode of interaction during his professional career. (4) The computer should not undertake all computational and graphic tasks. Accordingly, almost all of the computer programs of Project CAPLIN do require that students carry out either rough calculations, some part of the graphing, check hand calculated results with computer results or combinations of these. It is in fact the interaction of such student participation plus accurate, fast, computer calculated results which we find to be a tutorial benefit. (5) Material developed should be readily usable by others in the field. Accordingly, all material has been developed using standard computer languages and commercially available computer components.

In keeping with the fact that the students are not science majors the various computer tasks are all preprogrammed. Students are, however, able to assemble those parts of the program they choose to use. In addition, to within the logical requirements of the program, the students are able to skip randomly among various sections of the program, and if necessary, return to earlier sections for revision, etc. Several error recognition routines have been incorporated into the computer programming to insure that the student's interaction with the computer is as free of frustrations as possible.

These features and others are discussed below.

Equipment

The computer equipment available within the laboratory consists of Model 33 ASR teletypes coupled to external telephone lines via Model 113A Dataphones. Each teletype is linked to a Model 212 XY plotter manufactured by the Time Share Peripherals Corporation[1]. Three teletype-plotter units are available in each laboratory class to serve normally a total of twelve experimental teams of two students each. Within the confines of the two-hour laboratory session each team is allotted approximately 10-15 minutes of computer time. The equipment has been tested on an average of 300 students per week. With minor exceptions, the equipment has been found to be serviceable, rugged, though somewhat slow in information transmission (110 baud).

Currently the terminals, described above, have been linked to the time-sharing facilities of the Com-Share Corporation at Ann Arbor, Michigan. Programming of computer material has been carried out using XTRAN which is an extended FORTRAN language available on the Com-Share IDS 940 system. Recently, we have also transcribed all material into FORTRAN for operation on the CDC 6600 Intercom system of Indiana University's Research Computer Center. In so doing, we have had to write or adopt various intermediate level programs to simulate features normally available under XTRAN. In particular, these include syntax error detection, escape, plotting, and free formatting routines. For the last named we have adopted the programs reported by Smith[2]. Conversion was readily carried out within a period of roughly one month.

Although the equipment has served well the initial requirements of the project, it possesses two easily recognizable disadvantages: slow rate of information transfer and consequent high cost of computation. The initial average cost on the commercial time-share system was approximately \$2 per student per laboratory (not including terminal costs).

A major bottleneck is the 110 baud rate of the teletype terminal. On the other hand, we find that students appreciate the availability of hard-copy teletype and plotter output for future referral and for inclusion in the laboratory reports.

In order to economize it was necessary to shift some terminal instructions to mimeographed hand-out sheets, and to curtail the scope of some calculations, plots, simulations, etc. Many of these features could be reinserted providing a faster, inexpensive terminal were available. Toward this end we have briefly experimented with a Tektronix Model 4002 graphic display unit. Although such CRT display terminals provide the possibility of more rapid information transfer, they appear to be of little use in our environment unless hard-copy capabilities are included.

Programmed Laboratories

General Features. A total of 12 laboratories have been programmed to date. The individual programs will be listed in some detail in the next section. First we discuss general features common to all programs. These common features are incorporated into subroutines available upon call from a common CAPLIN library. Among the major of such common features are the following:

1. A sign-in and sign-out procedure which serves to initiate and end the program for the student, identifies the student and determines the elapsed time of each usage of the program.
2. A syntax error detection routine is applied to each line of entry carried out by the student. A message is given describing the nature of any error and a request for re-entry of the last line. This feature is particularly important when long data lists must be entered. Since many non-science majors are not adept at numerical manipulations, line by line error detection helps to alleviate student frustration during terminal interaction.
3. A data acquisition routine allows presetting of minimum, maximum, as well as fixed number of required data entries. This routine is particularly useful for entry of data in tabular form. The routine automatically requests entries until the preset number of entries per line and number of lines is satisfied. If the total number of data entries is to be fixed by the student, the routine counts the number of such entries and returns that number to the main program.
4. All programs are subdivided into parts. These parts may include data entry, error analysis, data reduction, tabular display, graphic display, and simulation sections. A routine called CHOICE permits the student (within the logic of the total program) to decide whether or not to use any particular part. If the student decides not to use a particular part, he is required to call up a new part. The latter feature permits random access (again within the logic of the total program) to any preceding and/or following parts. As an example, a particular student may choose to obtain the tabular display of the reduced data, but to skip over the plotting routine and carry out the latter on his own.
5. During execution of any of the parts, the student may also switch to any of the other parts of the program through the ESCAPE feature. This feature is accessed by depressing the escape key, or typing ESC, on the teletype, which automatically returns the program to the CHOICE subroutine described in the preceding paragraph.
6. A number of plotting packages have been developed with which the program author can, with a very small number of calling statements, produce graphic display of the experimental data and calculated results. One such plotting package called GENPLOT is also available to the student for the plotting of an array of X-Y points either directly or with a least-squares fit to the points.

Computer Aided Laboratories. The following is a list of the main features of each of the computer aided physics laboratories programmed and tested to date. An error analysis of the data is normally available in the following laboratories, even where it is not specifically mentioned in the description.

1. Computer familiarization. A single ten minute session is required to familiarize the student with the mechanics of the teletype and the plotter. The program also introduces the student to error messages, the CHOICE routine, data entry in tabular form, and plotting of graphs.
2. Measurements laboratory. Nominally this laboratory involves the measurement of the dimensions and density of a number of objects and with a variety of instruments. In actuality the laboratory is used to instruct the students in the concepts of data error analysis. The meaning of average value, mean deviation, and percent error are

illustrated. Most subsequent laboratories include error analysis routines which are optionally available. The teaching of error analysis and data value and the relatively rapid understanding of the subject by the students has been one of the major tutorial benefits of this project.

3. **Concurrent Forces.** The laboratory uses a simple force table consisting of two or more weights hung by strings to a central ring. The strings pass over roller-bearing mounted pulleys. The object of the laboratory is to find experimentally the resultant of three forces in equilibrium, to obtain the force of friction of pulleys on the force table, and to determine the direction and magnitude of an unknown force. The computer is programmed to calculate the resultant of three forces which the student claims to have so placed on the force table as to be in equilibrium. The student must, however, first enter his value for the resultant force. The computer then prints out its value as a comparison.

Next a computer aided exercise is performed to determine the magnitude and direction of an unknown force required to balance three known forces. In an optional exercise the force of friction is determined. The student finds the slope of a graph of the weight of the hanging masses vs. the incremental weight required to bring the system out of equilibrium. The results are checked against those obtained by the computer using a least square analysis of the data. A computer based plot is also available.

4. **Uniform Acceleration.** The laboratory involves the rather traditional experiment of the measurement of the acceleration of gravity by studying the motion of a body along an inclined linear air track[3]. Progress of the moving body down the track is recorded on a strip of spark sensitive tape. Before entering data into the computer, students are urged to scan their data to ascertain that the second differences of the position of the moving body as a function of time are roughly constant. The computer program also carries out the above, and if a preset number of second differences exceed a preset deviation from the average value, then the student's data is rejected, and a teletype graphic display of the data points is given.

If the data points are acceptable the computer program can calculate incremental velocities and acceleration, an error analysis of the acceleration, the acceleration of gravity, and the percentage differences between the accepted value and the experimentally obtained value of the latter. The results are available in tabular and graphical form.

5. **Newton's laws.** Apparatus for this experiment closely follows that of the experiment on uniform acceleration. Spark tapes are made of four runs during which the accelerating force (a hanging weight) on the moving body is incrementally increased while keeping the total mass of the accelerating body and hanging weight constant. The experimental data leads to the determination of the acceleration of the moving body under various forces, the acceleration of gravity and a check of Newton's force law. The computer scans the data from each of the four sparker tapes for constant second differences. If the data is acceptable the velocity of the body as a function of time is determined and displayed in either (or both) tabular or graphic form. A graphic display of the acceleration as a function of the mass of the accelerating force is also available. The latter leads directly to the determination of the acceleration of gravity, which is left to the student to calculate from either the tabular or graphical displays.

6. **Momentum Ballistics.** Studies the law of conservation of momentum and the elements of projectile motion. Specifically the initial velocity of a projectile is determined by two methods: (1) through a measurement of its range and vertical distance of flight and (2) through a measurement of the displacement of a ballistic pendulum. In the experiment a spring-gun ballistic pendulum is used as apparatus[4].

The computer is programmed to carry out an error analysis of the experimentally obtained range of the projectile motion, and of the maximum height of the ballistic pendulum. The system then calculates the initial velocity of the projectile based on the two methods and the percent difference.

A further check on the equations of motion is accomplished as follows. The initial velocity of the projectile is varied by changing the tension of the spring in the spring-guns among the apparatuses throughout the laboratory. The projectile range as a function of the initial velocity, as obtained by the various experimental teams, is then plotted on the computer system by the laboratory instructor.

A variation in the experiment is possible by using the computer to calculate the expected range of the projectile, based on the results of the pendulum. The student then returns to the apparatus to check the validity of the prediction.

7. **Uniform Circular Motion.** Laboratory to determine the centripetal force and acceleration of a body undergoing uniform circular motion. A centripetal force apparatus mounted on a variable speed rotator is used[4]. In this apparatus a spring (attached to the rotating body) provides the centripetal force. The latter is calculated by the computer from the experimentally determined rotational frequency, mass of the body, and radius of circular motion. The computed value is checked against the value calculated by hand by the student. A similar determination is made for the static gravitational force required to cause the same extension of the spring as in the dynamic part of the experiment. A computer based optional simulation experiment is available. The experiment involves graphed results of circular orbital motion of a satellite or planet. The student is free to fix the orbital velocity of the body, and the gravitational mass.
8. **Torsional Pendulum.** A laboratory to determine the period of torsional vibration, the factors which determine the period of vibration, the torsion constant of a thin rod, and the rotational inertia of various objects[4]. The computer is programmed to calculate the rotational inertia of discs, cylinders, rings and various combinations, rotating about a rod passing through their axes and also about displaced axes. In addition it determines the period of vibration and the torsion constant of the support rod. Optionally available is a calculation of the torsional modulus of the material of the rod.
9. **Two-Dimensional Collisions; Scattering.** Experimental apparatus consists of a hidden cylindro-polygonal object at which the student projects small steel balls[5]. The scattered projectile leaves marks on pressure sensitive paper which rings the unknown object. Students determine the angular location of the maxima in the scattering distributions. From these it is possible to determine the shape of the polygon and its orientation with respect to the trajectory of the projectile. The computer is programmed to check the students' determinations. However, the student is forced to determine the number of sides of the polygon before the computer becomes available to him. An error analysis is optionally available, as is a determination of the dimensions of the object providing the angular location of the "shadow" is supplied. The "shadow" is the location of the scattered distribution of projectiles which just miss the unknown objects on two sides.
10. **Simple Harmonic Motion.** Apparatus consists of two springs and various weights. The aim of this experiment is to obtain the spring constants of each spring and that of the springs combined in series. Two methods are used; (1) measurement of the period of harmonic motion for various weights, and (2) static displacement of the spring for various weights. The computer system is optionally available for plotting of data, and for least square determination of the spring constants using data from the two methods.
11. **Wave Motion.** Aim of the experiment is to check the equation of wave motion for standing waves of a vibrating string. Apparatus consists of a string held in tension by a weight and forced to vibrate by one side of an electrically driven tuning fork[4]. The string is "essentially" fixed at one end by the tuning fork and at the other end by a pulley. The student measures the density of the string and the weight required to bring the string to vibrate in its fundamental mode, and various harmonics. The computer is programmed to calculate the tension in the string, the velocity of propagation of the displacement, the wavelength, and frequency of vibration. A check is available on the experimental masses through a calculation of the mass required to obtain the above results. A plot of velocity vs. wavelength is also available for a graphical determination of the vibrational frequency.
12. **Velocity of Sound.** Laboratory to determine the velocity of sound in air, and to study standing waves in a tube open at one end[6]. The apparatus consists of a glass tube in which a column of water can be adjusted to any desired height. The student measures the length of unfilled tube required to fulfill the fundamental, and several harmonic, resonant conditions. This is done for a number of different sound frequencies (tuning forks). From the measurement the velocity of sound can be abstracted. The computer is programmed to carry out the calculation of the velocity of sound corrected for the temperature of the room. A computer simulated experiment is also available. In the latter, the computer plots out a graph of two sine waves differing in frequency by a small amount. It also plots the sum of the two sound waves, and its envelope to illustrate the beat frequency. The student is then required to return to the laboratory, obtain a second sound column, and check for the occurrence of the beat phenomenon.

Conclusion

A survey of student reactions to Project CAPLIN has been carried out by direct observation in the laboratories, by discussions, and by a formal questionnaire of 300 students involved. We do not claim, however, to have carried out a thoroughly conceived and tested statistical survey. Nevertheless we have gained some insight into the benefits of the project to the students and their reactions to it.

Perhaps the most easily recognizable benefit is the rapid rate at which the students learned to understand and appreciate the error analysis of experimental data. This result was achieved through consistent application of data error analysis as taught to the students during the measurements laboratory discussed in the previous section. Many students eagerly awaited the print out of the percent error in the data, and quite a number actually repeated experiments (in a knowledgeable manner) when significant error was indicated. Such results had not been possible prior to the implementation of the computer system. In the past, data error analysis was normally neglected in order to maintain a reasonable duration for any one laboratory. Even when such error analysis was included as a long hand exercise it was seldom possible for the student to rerun the experiment within the time limitation.

A second readily appreciated benefit of the project has been the tutorial aspect of computer based graphing. By providing the student with accurately plotted graphs at various phases of the program, and by requiring abstractions from or additions to the graph, it has been possible to help the student to become familiar with graphical representation of physical concepts. We have observed that many non-science majors have had only limited previous exposure to such knowledge. We are currently further refining the techniques used in student-graphics interactions.

Students also found it advantageous to their understanding of the experiment to have available accurately calculated results of some phases of the total data reduction. In almost all laboratories, student participation is required through additional calculations. The availability of some calculated results serves as a useful guide against which to measure their understanding and the accuracy of the results of the additional calculations.

Finally, the increased rate of comprehension in the areas discussed above, plus the inherent saving of time in calculations performed by computers, allowed us to expand a number of the laboratories. The expansion has taken the form of either increasing the duration or complexity of the experimental phase of the laboratory, or in carrying out simulated allied experimentation on the computer system. In this way, it has been possible not only to enrich the laboratory experience of the student, but also to cover additional physical principles and their extensions.

REFERENCES

1. Time Share Peripherals, Wilton, Connecticut.
2. Computer-Based Physics: An Anthology, Ed. R. Blum, Commission on College Physics, September 1969.
3. Manufactured by The Ealing Corporation, Cambridge, Massachusetts.
4. Manufactured by Central Scientific Company, Chicago, Illinois.
5. Manufactured by Welch Scientific Company, Skokie, Illinois
6. The experiment follows the discussion in Chapter 20 of "Physics" Part 1, by R. Resnik and D. Halliday, John Wiley and Sons, New York, 1966, page 509 ff.

A COMPUTER ASSISTED INSTRUCTION PROGRAM IN
AMERICAN HISTORY, 1870-1921

Betty L. Jehn
University of Dayton
Dayton, Ohio 45459
Telephone: (513) 434-4520

This instruction program is an attempt to demonstrate the adaptability and versatility of the computer in a field other than science, and by so doing to provide a new approach to the learning of American History. This program is designed as a study aid and hopefully it will enable the student to find deeper meaning and understanding from a textbook and classroom instruction.

In the past five years there has been a shift in the philosophy of education in the United States, but it seems that the technology of education has not kept pace, at least on the college level.

With the increasing awareness of disadvantaged groups in our society, our approach to education has shifted from establishing simply mass education to providing quality mass education[1]. One result of this new approach has been a change of administrative attitude on the college level from high-selectivity for admission to more opportunities for the less well-prepared students. Consequently, numerous problems have arisen among which are increasing enrollments and rising costs.

These increased costs; the need to educate more people from different backgrounds and of different ages; the fact that the body of knowledge in all disciplines is expanding even faster than the body of students; the fact that the teacher of tomorrow must present more information using the same period of time that he uses today; and the limited resources of equipment and personnel, all these demand some new approach to the solution of the problems of education. Educational technology and computers in particular hold forth a promise of that solution to the humanities as well as to the sciences.

As other types of education technology have proven less than phenomenally successful, educators have turned more and more to the computer. The computer has capabilities undreamed of even by the computer specialists of just a decade ago.

The history profession in recent years has been modernizing to the extent that ethnic studies, area studies and comparative studies are being incorporated into the courses. But the great problem in teaching is communicating with and motivating contemporary students.

Two young professors at Orange Coast College, Costa Mesa, California express the problem in this way:

At a time when people spend a greater portion of their lives before television tubes than in educational institutions, instructors lull students to sleep with lectures such as were designed for information transferral prior to the invention of the printing press. While the public is accustomed to sophisticated motivational psychology from Madison Avenue, instructors rely upon primitive reward-punishment mechanisms to interest students in learning. And while the public is effectively tied to instantaneous electronic data systems, students are asked to crowd their minds with raw data as if it contained some inherent meaning.

Such poor allocation of resources is grossly inefficient and really indefensible. Historians like others operating in the twentieth century, ought to exploit the knowledge, skills, and tools available to them for relating their discipline to their students[2].

Dr. Robert D. Hess, professor of psychology in Stanford University's school of education and a colleague, Dr. Maria D. Fenezakis have found that a group of junior high school students aged thirteen to fifteen associated the computer with such human qualities as trustworthiness, reliability, veracity, and fairness. The study further indicated that the youths rate the computer as a "more positive source of information" than their teachers, textbooks, or television news reports.

The use of technology in education and particularly that of the computer for instruction, will continue to expand. Thus the adaptations that the child develops in his responses to machines become of special interest.

These are crucial aspects of the socialization of the child into modes of dealing with an industrial highly technological society. The points of contact between the child and the institutions of society will be increasingly mechanical. Thus the machine takes on the role of an authority figure [3].

As a result of these observations, the decision was made at the University of Dayton, in the spring of 1970, to attempt a Computer Assisted Instruction program in American history as a master's thesis. As of that time, as far as we were able to ascertain, little or nothing had been done in the field of history in CAI. My rationale for choosing two seemingly far removed disciplines in which to work was manifold: I was working in American history; I needed something new and different to use as a thesis; I had an interest in computers and computer science; and the University of Dayton has an RCA Spectra 70/46 computer which has the capability of accommodating CAI. Everything seemed to point in one direction and the chairman of the history department was receptive to the idea so I went ahead.

When we began this CAI experiment in American history, it was decided to program it in BASIC. BASIC was chosen for two reasons: the programmers were familiar with it and it is compatible with the RCA Spectra 70/46. Two computer science students, Karen Rudzinski and Frances Gundy, did the programming for this course. It was used as their special project for a course in Advanced Programming.

As a basis for the questions in this program the American History Summary Cards, Set No. 2 written by Dr. Wilfred J. Steiner of the University of Dayton were used. The textbook by Samuel Morison and Henry Steele Commager, Growth of the American Republic, Vol. II, New York, 1969, was also used in order to verify the arrangement of the material. Any other text could be used and the material rearranged to fit. A fact is a fact and will be found in any reliable textbook.

This program in American History begins at the end of Reconstruction and ends at the Treaty of Versailles. It consists of eighteen separate parts or lessons. As each part requires a great deal of computer storage space, only the first seven were put on the system in the beginning. The remainder are stored on decks of IBM cards and can be put on the system at a moment's notice. Because of this physical storage problem, it is recommended that only a certain number be put on the system at a given time. For example, five lessons could be put on at the beginning of a term; at the end of a period of time when the instructor felt that the material had been covered in class; lesson one or perhaps one and two could be taken off and six or six and seven added. This is accomplished very quickly and easily. It is always possible to put the programs back on the system and perhaps a day or so before an exam, the entire group of lessons could be put on for further study.

By dividing the History Program into various segments, it becomes easier to handle than one continuous program. This is especially true if it is to be used on a system smaller than the RCA Spectra 70/46. Also by having small lessons, revisions and alterations can be very easily accomplished. Because the instructor himself can revise and alter the program, it becomes not a standardized, stereotyped instruction, but rather a course of study personalized by the individual instructor.

The goal of tutorial CAI is to individualize instruction and while interaction with a computer is more personal than for example, merely watching a film strip, nevertheless the personal quality is missing. CAI will never replace a classroom teacher, but assuming that the student will have access to most of the information that can be offered in a specific course from a variety of sources, one can readily understand that this availability of educational resources must bring about a change in the role of the instructor. The computer will relieve the teacher of many routine burdens thus giving him additional time and then the role of the teacher can change from one who merely imparts information to one who relates to the students, and by so doing becomes one who imparts knowledge by managing the learning experiences.

In the case of a CAI program in history, the computer can impart the facts and events of a certain period. But a group of facts is not history, however a student must have a background of events before these same events can be interpreted. A teacher, therefore can be relieved of the burden of fact presentation and can take the valuable classroom time for discussion, theories, and interpretations - a more in-depth coverage of the material would thus be possible in the same time allotment.

A program such as this can be of almost unlimited value. When the situation arises wherein there is a potentially good student but with a poor background in a class, the program could be used to give such a student a quick grasp of fundamental facts. It could serve as a review of fundamentals which an average student should know, but does not. It could serve as a fast review before a test. It could reinforce the learning process for a slow learner. With a few alterations, it could be used as a placement test for a transfer student or a student with a superior high school background. Further, university classes in the survey courses are

frequently large, this CAI program could supply the individual attention which would otherwise be impossible.

This particular CAI program in American History consists of a series of questions and answers. The student sits at a terminal and identifies himself to the computer by giving his student number. He is then asked which lesson he would like to do. After typing in the code name of a lesson on the terminal he is ready to begin. The computer will ask a question which will require the student to answer in one of several ways. Sometimes the question will be a simple true or false; sometimes it will be a multiple choice; and sometimes the student will be required to supply the answer himself. The computer will then tell the student if he is right or wrong, and if the answer is wrong the student will be given the correct answer and in many cases an explanation will be given.

A group of students from a University of Dayton history course were asked to use this program during the spring term of 1970. About half of the students in one particular course volunteered to use the CAI program. At the end of the term, these students were asked to fill in a questionnaire regarding the program. The results of the questionnaire were most encouraging. In every case the students using the CAI program considered it "fun." The following are some typical comments made by the students:

I really liked working the CAI program -- it would be a good use for review.

I thought the program was very helpful, and helped reveal one's weak and strong areas on his knowledge of the segment of history being covered.

I feel that an attempt was made to set up a history program that was stimulating, interesting and in general a fine attempt to bring technology to the student. As a participant I thank you for the opportunity to learn just a minute portion of a thing I have long held mysterious (The Computer)

I think that a history course using this type of program would be most worthwhile on the high school level. I don't think that facts should be so emphasized in the class work or exams of a college history course. I feel that analysis is more important on the college level. A good background of facts should be gained in high school, and I think high school students would find it to be a great help.

I think it could be a very good review for tests.

Just a lot of fun for a change! I really liked it and it proved how little I know in history.

If this type program were to be used for future classes as study material, I think it would be wise to have the students prepare some of the questions. The only criticism I have is that if your answer is incorrectly spelled, but the name is correct, it is not correct. But I would consider this a very minor criticism in comparison to the help I consider CAI does.

The students using this program seemed to be very concerned when the computer told them they were wrong. They were especially concerned if a question was counted incorrect if it was merely misspelled. When CAI is explained to students in the future it will be wise to explain that the computer accepts exactly what one tells it and not what one thinks he has told it. Also that the program is for study and not a test and that they should not be overly concerned if a misspelled word is not accepted. Some ordinary misspellings were included in the program as acceptable, but it is impossible to anticipate all the variety of ways a word can be spelled! For this reason then, if this history program were to be used as a test, it is recommended that questions be of true and false or multiple choice only to avoid the previously mentioned possibilities or rejection for misspelling.

We have all accepted the premise that knowledge is ever increasing and at a fantastic speed, therefore our ways of imparting knowledge must necessarily change. An effective learning experience is one in which a student will receive multiple exposure to the material. There is nothing in a CAI program which could not be found in a textbook, but it is presented in such a completely different way that it makes a different impression on a student. When the material necessary for the achievement of a learning objective is presented through a variety of media, the probability that the student will get the multiple exposure necessary is greatly increased. And because of the interaction of the student with the computer, it is a way of getting something more than passive acceptance from a student.

Because the University of Dayton was among the first universities in the nation to have a computer facility, I feel that it should also be in the forefront of computer usage. Since computers are such a tremendous investment, they should be used in every conceivable way to make the investment profitable.

0LO
OLO PROGRAM NAME--LABOR01
READY

*RUN
WHAT'S YOUR NICKNAME?BETTY
WHAT IS YOUR STUDENT NUMBER ?
712345

ANSWER 'TRUE' OR 'FALSE' TO ALL TRUE OR FALSE QUESTIONS
UNLESS GIVEN OTHER DIRECTIONS. ALSO GIVE ONLY THE LAST
NAME WHEN A FILL-IN REQUIRES A PERSON'S NAME.

1 . IN 1900 OVER ONE AND A HALF MILLION CHILDREN UNDER 16
WERE EMPLOYED IN FACTORY OR FIELD IN THE U.S.

?TRUE
I'M PROUD OF YOU, BETTY

2 . CHILD LABOR LAWS AROUND 1900 APPLIED ONLY TO CHILDREN
UNDER ----- YEARS OF AGE. (USE NUMBERS)

?12
60=0

3 . INDUSTRIAL ACCIDENTS WERE COMMONPLACE.

?1
NO, NO= NO, BETTY

4 . THE KNIGHTS OF ST. CRISPIN WAS A UNION OF:
?RAILROAD WORKERS

WRONG
IT WAS A UNION OF SHOEMAKERS ORGANIZED IN 1867.
IT USED REACTIONARY MEASURES AND WAS NOT SUCCESSFUL.

5 . THE FOUR LARGEST RAILROAD UNIONS WHICH ARE STILL
OPERATING TODAY ARE KNOWN AS THE A) RAILROAD BROTHERHOODS
B) RAILWAY LABOR UNIONS

?A
RIGHT

6 . SAMUEL GOMPERS AND ADOLPH STRASSER ORGANIZED WHAT
GROUP IN 1886 AT COLUMBUS, OHIO? (USE INITIALS
AND NO PERIODS)?AFL
THAT'S RIGHT

7 . WHO WAS THE FIRST AND ONE OF THE MOST IMPORTANT
PRESIDENTS OF THE AFL?GOMPERS
HE WAS PRESIDENT FROM 1886 UNTIL 1924.

8 . WHO TOOK OVER THE PRESIDENCY OF THE AFL IN 1924
AFTER GOMPERS?SORRY
THAT WAS A LITTLE ROUGH. HE WAS WILLIAM GREEN.

9 . TERENCE V. POWERSLY WAS GRAND MASTER WORKMAN OF THE
ORGANIZATION CALLED THE:

?CIO
THE ORGANIZATION WAS CALLED THE KNIGHTS OF LABOR.

10 . THE WAGE PRINCIPLE ESPOUSED BY DAVID RICARDO THAT ALL
WAGES TEND TO FALL TO THE LEVEL WHICH THE MOST UNSKILLED
OR THE MOST DESPERATE MAN WOULD ACCEPT IS:

A) SUPPLY AND DEMAND LAW
B) IRON LAW OR WAGES
C) NEGATIVE WAGE PRINCIPLE

?B
GLAD TO SEE YOU GOT THE LAST ONE RIGHT!
YOUR SCORE IS 60 %
YOU DON'T KNOW THIS TOPIC ADEQUATELY.

*BYE
BKPT POINT 006572

/LOGOFF

XC E420 LOGOFF AT 1500 ON 02/26/72, FOR TSM 3777.

XC E421 CPU TIME (SEC.) 2026.4133 CONNECT TIME (MIN.) 023

XC E427 JUS: SPENT \$1.92 SPENT TO DATE \$220.61

FIGURE 1. Sample run of one lesson in the program: Labor.

QLO PROGRAM NAME--IMMIG
READY

*RUN

HI! MY NICKNAME IS SPECTRA. WHAT'S YOURS?BETTY

WHAT IS YOUR STUDENT NUMBER?12345

PERSISTANT POVERTY FOR THE PEASANTS; RECURRENT HARD-TIMES FOR WORKERS; WAR AND THREAT OF MILITARY SERVICE; POLITICAL OPPRESSION; RELIGIOUS PERSECUTION AND CLASS-SYSTEM WHICH CLOS=O DOORS TO OPPORTUNITY WERE MAJOR MOTIVATIONS FOR ENIGRATION TO AMERICA.

- 1 • NAME TWO OTHER COUNTRIES BESIDES THE UNITED STATES WHICH RECEIVED MANY IMMIGRANTS DURING THE NINETEENTH CENTURY. (ENTER ONE COUNTRY PER LINE.)

?MEXICO

?CANADA

CANADA IS CORRECT BUT MEXICO IS WRONG.
OTHER COUNTRIES RECEIVING MANY IMMIGRANTS WERE:
AUSTRALIA
SOUTH AFRICA
ARGENTINA
BRAZIL

- 2 • C=CAP IMMIGRANT LABOR THREATENED THE GAINS OF ORGANIZED LABOR WHICH CAUSED UNIONS TO AGITATE FOR ? OF IMMIGRATION.

?QUOTAS

CORRECT.

- 3 • IN THE DECADE FROM 1850 TO 1860 TWO AND A HALF MILLION IMMIGRANTS CAME TO AMERICA. (ANSWER TRUE OR FALSE.)

?TRUE

YOUR ANSWER IS RIGHT. ANOTHER 14 MILLION ENTERED FROM 1860 TO 1900.

- 4 • IN THE GENERATION IMMEDIATELY AFTER THE CIVIL WAR, A LARGE NUMBER OF IMMIGRANTS TO AMERICA CAME FROM:

- A. NORTHERN AND WESTERN EUROPE
B. SOUTHERN EUROPE

?A

VERY GOOD.

- 5 • THE IRISH IMMIGRANTS TENDED TO:

- A. BECOME FARMERS
B. SETTLE IN THE CITIES

?A

WRONG. THE IRISH MADE LOUSY FARMERS.

- 6 • RAILROADS AND STEAMSHIP COMPANIES ENCOURAGED IMMIGRATION BY THEIR ADVERTISING IN EUROPE. (TRUE OR FALSE)

?FALSE

HOW DO YOU THINK ALL THOSE POLLOCKS GOT OVER HERE ?
SAY, THAT REMINDS ME OF A GOOD POLLOCK JOKE: WHAT'S A POLLOCK'S IDEA OF MATCHED LUGGAGE?.....TWO A&P SHOPPING BAGS!!!

FIGURE 2. Sample run of one lesson in the program: Immigration.

- 7 • LARGE NUMBERS OF GERMAN IMMIGRANTS CONGREGATED
IN THE ? TO BECOME FARMERS.
?MIDDLE WEST
CORRECT.
- 8 • LILLIAN D. WALD AND JANE ADDAMS WERE EARLY CHAMPIONS OF:
A. SOCIAL REFORMS
B. POLITICAL REFORMS
?B
YOU HAD A 50-50 CHANCE OF GETTING THIS RIGHT AND
YOU BLEW IT!
- 9 • DURING THE LATE NINETEENTH CENTURY AND EARLY TWENTIETH
CENTURY, AS HIGH AS 75% OF MINE LABORERS WERE
FOREIGN BORN. (TRUE OR FALSE)
?TRUE
IT'S A GOOD THING YOU GOT THIS ONE RIGHT.
- 10 • THE IMMIGRATION ACT OF 1902 PROVIDED FOR A LIMITED
NUMBER OF IMMIGRANTS FROM SOUTHERN EUROPE (T OR F)
?F
WRONG. THIS ACT PROHIBITED THE IMMIGRATION OF CRIMINALS
AND OTHERS LIKELY TO BECOME A PUBLIC CHARGE.
- 11 • AN ACT WHICH PROHIBITED THE ENTRY OF CHINESE LABORERS
INTO THE U.S. FOR A PERIOD OF TEN YEARS WAS --- --- ACT.
?I DONT KNOW
BETTY, YOU SHOULD HAVE KNOWN THIS ONE!
THE ACT WAS THE CHINESE EXCLUSION ACT.
- 12 • IMMIGRANTS FROM WHICH COUNTRY OF WESTERN HEMISPHERE
CAUSED THE MOST PROBLEMS TO THE U.S. (MEXICO OR CANADA)
?MEXICO
MEXICO IS CORRECT.
- 13 • THE DIFFERENCES BETWEEN THE 'OLD' IMMIGRANTS AND
THE 'NEW' IMMIGRANTS WAS STRIKING. DID THE DIFFERENCES
DISAPPEAR WITH THE SECOND GENERATION? (YES OR NO)
?NO
WRONG. THEY DID DISAPPEAR WITH THE SECOND GENERATION.
- THE NUMBER WRONG = 7
YOUR SCORE IS 46 %.
ARE YOU GOING TO LET A COMPUTER OUT-SMART YOU?
TRY STUDYING THIS SHEET.
•

FIGURE 2. Continued.

The President's Science Advisory Committee Report on "Computers in Higher Education" makes abundantly clear the necessity of quickly introducing computer instruction and time-sharing computer experience into universities, colleges, and high schools. The Report says, "Undergraduate college education without adequate computing is deficient education, just as undergraduate education without adequate libraries would be deficient education."

A personal hands-on acquaintance with a computer terminal should be part of the experience of every college student.

FOOTNOTES

1. Lawrence P. Grayson, "A Paradox: The Promises and Pitfalls of CAI," Educom, V, No. 2 (March 1970), 8.
2. Michael G. Crow and Edward A. Burke, Jr., "Modern Methods can Make History Relevant, Interesting," Educational Media, VIII (Feb. 1970) 8.
3. Harvey Elman, "These Students Trust Computers, Call Them Reliable," Computerworld, (May 6, 1970), 25.

REFERENCES

BOOKS

1. Basic Language, Time-Sharing Service Reference Manual, General Electric Information Service Department, June 1965.
2. Bushnell, Don D. and Dwight W. Allen, The Computer in American Education, New York, 1967.
3. Caffrey, John and Mosmann, Charles J. Computers on Campus: A Report to the President on Their Use and Management, American Council on Education, Washington, D. C. 1967.
4. Pink, Donald, Computers and the Human Mind: An Introduction to Artificial Intelligence, New York, 1966.
5. Lekan, Helen A., ed, Index to Computer Assisted Instruction, Instructional Media Laboratory, The University of Wisconsin-Milwaukee, Second Edition, Sterling Institute, Feb., 1970.
6. Morison, Samuel and Henry Steele Commager, Growth of the American Republic, 2 Vols., New York, 1969.

ARTICLES IN PERIODICALS

1. Anderson, G. Ernest, "Computer-Assisted Instruction: State of the Arts." Nation's Schools, LXXXII, (October, 1968), 50-54.
2. Blaschke, Charles, "Computers in Education: Interesting, But How Relevant?" Educational Technology, IX (May 1969), 5-8.
3. Carson, Linda, "CAI--What It Can Do for Education" Educational Media, I, (October 1969), 6-8.
4. Crow, Michael G. and Edward A. Burke, "Modern Methods Make History Relevant, Interesting" Educational Media, I, (February 1970), 8-10.
5. Elman, Harvey, "These Students Trust Computers, Call Them Reliable," Computerworld (May 6, 1970), 25.
6. Feingold, Samuel L., "Planit--A Language for CAI," Datanation, XIV (September 1968), 150-154.
7. Fryo, Charles H., "CAI Languages: Capabilities and Applications," Datanation, XIV (September 1968), 140-145.
8. Gilman, David Alan and Nancy Ann Morean, "Effects of Reducing Verbal Content in Computer-Assisted Instruction." A V Communication Review, XVII, (Fall 1969), 16-20.

9. Grayson, Lawrence P., "A Paradox: The Promises and Pitfalls of CAI," Educom, V (March 1970), 1-3
10. Harman, Don, "Cincinnati Students Vie For Computer Time," Educational Media, I (September 1969), 4-7.
11. Inman, Richard P., "Computer-Assisted Education at the Naval Academy," Educom, IV, (March 1969), 1-4.
12. Kanner, Joseph H., "CAI--The New Demonology?" Datamation, XIV (September 1968), 151-154.
13. Kantor, Seth, "Blueprint for Revoultion," Educational Media, II (April 1970), 3-6.
14. Luskin, Bernard J., "Computer's Educational Role Confused by the Lack of Standard Definitions." Computerworld, January 15, 1969.
15. Luskin, Bernard J., "The Time is Now for Development of Computer Assisted Learning," Computerworld, November 17, 1968.
16. Menkhaus, Edwin J., "The Computer Kids: Most Likely to Succeed" Computer Digest, (October 1968), 1-2.
17. Morrison, H. W. and E. M. Adams, "Pilot Study of a CAI Laboratory in German," The Modern Language Journal, LII, (May 1968), 10-12.
18. Oettinger, Anthony and Sema Marks, "Educational Technology: New Myths and Old Realities," Harvard Educational Review, XXXVIII (Fall 1968).
19. Riedesal, C. Alan and Marilyn N. Suydam, "Computer-Assisted Instruction: Implications for Teacher Education," The Arithmetic Teacher, (January 1967), 1-10.
20. Rogers, James L., "Current Problems in CAI," Datamation, XIV (September 1968), 130-133.
21. Smith, J. Stanford, "The Growing Maturity of the Computer Age," Computer Digest (November 1968), 3.
22. Suppes, Patrick and Mona Morningstar, "Computer-Assisted Instruction," Science, CLXVI, No. 3903 (October 17, 1969), 348.
23. Vannan, Donald A., "Educational Media in the U.S." Educational Media, II (April 1970), 7-8.
24. Zinn, Karl L., "Instructional Uses of Interactive Computer Systems," Datamation, XIV, (September 1968), 145-150.

UNPUBLISHED REPORTS
U. S. GOVERNMENT DOCUMENTS

1. Alcorn, Bruce K., "The Small College and the Computer--An NSF Experiment," Report on a National Science Foundation Grant. SREB, 1969.
2. Becker, James W., "Run Computer Run: A Critique," Presentation for the Conference on Information Technology and Secondary Education, May 1-2, 1968. Sponsored by Harvard University's Program on Technology and Society.
3. "Computer-Assisted Instruction at the Florida State University, IBM, Application Brief.
4. "Computer-Assisted Instruction, North/South/East/West," The Proceedings of Four Regional Conferences of ENTELEK CAI/CMI Information Exchange. 1969. Entelek Incorporated Newburyport, Mass.
5. Entelek, CAI.CMI Newsletter" November 1969
6. Hickey, Albert E., et al., "Computer-Assisted Instruction, A survey of the Literature, Entelek, Incorporated, Newburyport, Mass., Defense Documentation Center. Defense Supply Agency.
7. Kopstein, Felix P., et al., Computer-Administered Instruction Versus Traditionally Administered Instruction: Economics, Georgia Washington University, Alexandria, June 1967. Defense Documentation Center. Defense Supply Agency.

8. Majer, Kenneth and Duncan Hansen, "A Study of Computer-Assisted, Multi-Media Instruction" augmented by paper presented at the American Education Research Association Annual Meeting, Los Angeles, California, February 1969.
9. Marcus, Robert L., "Summary Report," Entelek CAI/CHI Information Exchange, Midwest Regional Conference, 1969-70.
10. Nevinson, John M., "Demonstration and Experimentation in Computer Training and Use in Secondary Schools." Secondary School Project Interim Report (No. 3). October 1, 1968.
11. Plyter, Norman, "CAI Program of the State University College at Brockport, N.Y." Summary Report, October 3, 1969.
12. Richardson, Jesse O., "Modern Trends in Education, One-Year Schoolwide Project, Grades K-12, Computers in the Classroom." Science Research Associates, Inc., March 15, 1968.
13. Richardson, Jesse O., "Programmed Instruction and Learning Systems, One Year Schoolwide Project," Science Research Associates, Inc., Chicago, April 15, 1968.
14. Rosenberg, R. C., et al, "Investigations in Computer-Aided Instruction and Computer-Aided Controls." USAF Bedford, Massachusetts, April 1967. Defense Documentation Center, Defense Supply Agency.
15. Seidel, Robert J., "Computers in Education: The Copernican Revolution in Education System." Professional Paper 16-69, May 1969.
16. Silverman, Harry P., "Applications of Computers in Education." Systems Development Corporation, Santa Monica, California, August 29, 1967.
17. Smallwood, Richard D., et al, "Quantitative Methods in Computer-Directed Teaching Systems." Stanford University, Stanford, California. Defense Documentation Center. Defense Supply Agency. March 15, 1967.
18. Strumm, Irene, "Use of CAI in New York State Public Schools." Summary Report, Board of Cooperative Educational Services #1. Yorktown Heights, N.Y. October 1968.
19. Suppes, Patrick and Mona Morningstar, "Evaluation of Three Computer-Assisted Instruction Programs," Technical Report, No. 142. Institute for Mathematical Studies in the Social Sciences, Stanford University, May 1969.
20. Zinn, Karl L., "An evaluative Review of Uses of Computers in Instruction Project CLUE (Computer Learning Under Evaluation) U.S. Office of Education, January 1, 1969 - January 31, 1970.

SPECIAL LISTING

1. Steiner, Wilfred J., American History, Summary Cards. Civil War - Present Time. Set No. 2, Visual Education Association, Inc., Dayton, Ohio.

A PRELIMINARY REPORT ON COMPUTER ASSISTED LEARNING
IN AMERICAN HISTORY COURSES AT OKLAHOMA STATE UNIVERSITY

Charles M. Dollar
Oklahoma State University
Stillwater, Oklahoma 74074
Telephone: (405) 372-6211

Introduction

Dr. Duane C. Spriestersbach, Vice-President for Research and Dean of the Graduate College at the University of Iowa, opened the 1970 Conference on Computers in the Undergraduate Curricula with an address in which he called for accountability by those urging further use of computers in higher education[1]. He suggested that the extraordinary momentum underlying computer usage, which was attributed primarily to federal funding, would diminish as budgetary constraints became more acute. This situation he warned, required that advocates of the use of computers in instruction in higher education take a hard look at how they are being used and collect hard data that document the difference, if any, they make in what students learn and how they learn. Toward this end, Dr. Spriestersbach proposed a list of questions that proponents of the use of computers in undergraduate instruction must face. If the papers presented at the 1971 Conference on Computers in the Undergraduate Curricula accurately reflect what is going on in the use of computers in higher education, then it is fair to say that Dr. Spriestersbach's exhortation has yet to be heeded seriously[2]. Accountability is still not a major concern.

This paper reports preliminary results of a program of computer assisted learning in American history courses at Oklahoma State University that attempts to come to grips with Dr. Spriestersbach's call for accountability. In no way is this paper to be construed as fully satisfying this call. Rather, it marks only a beginning. It is anticipated that the program at Oklahoma State University along with comparable programs at other institutions of higher education will generate the data required to assess the contribution computers make to learning[3].

Section I

The program itself, which is being funded by grants from the National Endowment for the Humanities, the Gulf Oil Foundation, and the Oklahoma State University Education and Research Foundation, will be discussed later. At this point it is necessary to consider briefly the development of the program.

Several years ago the writer became interested in exploring the possibilities of using a computer in teaching undergraduate American history courses. A review of the then available literature disclosed that computer usage generally consisted of drill and practice exercises, mastery of mathematical formulae and statistical techniques, and problem solving. None of these applications seemed particularly useful in undergraduate American history courses. Drill and practice exercises replicate rote memorization of factual information. Most students who enroll in American history courses have a minimal (or perhaps none) mathematical and statistical background. Furthermore, student mastery of the course content rather than statistical techniques for which they subsequently would have little use was a major concern. Use of a computer in teaching undergraduate American history, therefore, seemed inappropriate.

Further study of the problem, however, suggested that this dissatisfaction was unwarranted. Computer capability could be tailored to fit history courses rather than reshaping the courses to conform to the computer or dismissing it as irrelevant to undergraduate American history. Therefore, over a period of time several guidelines for adapting computer capability for use in teaching undergraduate American history courses evolved. They were: (1) use of a computer must be based on an explicit instructional strategy; (2) use of a computer must be clearly related to the content of the course and conventional instructional techniques; (3) use of a computer must concentrate upon information retrieval and simple arithmetic computations; (4) use of a computer must minimize student involvement in computer processing operations; and (5) use of a computer must be simple and straightforward enough to be understood readily by all students in the course.

These five guidelines were followed in constructing a series of computer assisted learning units for an upper division American history course. A major concern was the selection of an explicit instructional strategy. The inquiry approach of the "New Social Studies" was chosen because it focused upon learning rather than teaching and it emphasized the development of critical thinking abilities in a historical context which perhaps could be enhanced through computer technology[4]. There are five steps in the inquiry approach:

1. defining a problem
2. developing a testable hypothesis in terms of the problem
3. selecting the appropriate data for testing the hypothesis
4. testing the hypothesis
5. evaluating the results

Each of the computer assisted learning units, therefore, was structured in terms of the goals and activities of the inquiry approach.

Each unit was an integral part of the course and permitted students to examine a topic in greater depth than attempted in either the textbook or lectures. For example, one unit dealt with restrictive immigration legislation in the 1920's. The lecture and textbook context for this unit was intolerance in the 1920's, of which restrictive immigration legislation was one manifestation. Students could use lectures, textbook assignments, and special reading assignments as sources for defining some aspect of the problem of restrictive immigration legislation and developing a testable hypothesis about it. Thus, this unit, along with the other units, was a viable part of the course content and fit in with the traditional lecture approach.

The units were designed so as to require the use of a computer to retrieve information and to perform simple arithmetic computations. A data file appropriate for each unit was generated. The evidence could be retrieved as raw data or aggregated, subtracted, divided, or multiplied. One of the units focused upon voting patterns in New York, Michigan, and Nebraska from 1920 through 1940. The data file for each state contained county level units of information on some 300 variables. These variables included election returns for major elections and census information for each county for 1920- 1930, and 1940[5]. Thus, for each state there were about 20,000 separate items of information which could be retrieved or converted into another form such as percentages. For example, in testing an hypothesis about voting behavior in the presidential election of 1936 a student might want to compare those counties with a large Democratic plurality with those counties whose population was largely of foreign born extraction. Since much of the New Deal literature emphasizes the immigrant vote in that election, a student could hypothesize that those counties with a high percentage of foreign born stock would also be those counties with large Democratic pluralities.

Early in the developmental stage it became apparent that few students could be expected to submit a unique computer run which retrieved the desired information and performed specified computations. Accordingly, student involvement in computer processing operations was minimized by making available a complete listing of the raw data and all possible computations. This usually resulted in inundating students with information for which they had little use. However, in comparison to the problems encountered with each student attempting to submit a unique computer run, this inconvenience was minor.

The fifth guideline followed in developing computer assisted learning units was simplicity. Learning activity packages were designed which permitted each student to complete a unit at his desired pace (within certain limits, of course) with minimal dependence upon the instructor. Included in each package was a flow chart of required activities, student performance objectives, a review of the inquiry approach, a description of each variable in the data file, and an explanation of how to interpret the computer output. This emphasis upon simplicity did not confine students to simpleminded questions and problems. On the contrary, each learning activity package had an open-ended structure which encouraged students to develop and test significant hypotheses.

Section II

In the Spring of 1970 the Oklahoma State University Computer Center made available time-sharing conversational capability to the campus. This on-line interactive system, which is run under IBM's Conversational Programming System (CPS)[6], permits students to access data files via remote terminals and to retrieve information or to perform specified calculations. With the support of the National Endowment for the Humanities and the Gulf Oil Foundation the existing computer assisted learning units were converted to CPS and new units begun. A two year program was commenced which would: (1) measure the effectiveness of the "inquiry approach," supported by computer technology, in enhancing critical thinking abilities; (2) disclose students' perceptions of the function of computers in the instructional process without regard to discipline; and (3) determine the relative cost per student of offering this kind of learning environment. The balance of the paper describes the current status of the program and reports preliminary results regarding costs and students' perceptions. In addition, it documents some error and mistakes.

There are now six operational computer assisted learning units being employed in three American history courses at Oklahoma State University. The first of these, which is called RCVA for Roll Call Vote Analyzer, provides an on-line retrieval capability for the analysis of senatorial roll call votes[7]. The data file for RCVA will accommodate 100 senators' responses

of yea and nay to 25 roll call votes. The retrieval phase of RCVA offers students seven modes of access to the roll call votes. They are:

1. List a group of senators' responses to a particular roll call vote.
2. List a group of roll call responses made by a particular senator.
3. List a group of roll call responses made by the senators from a particular state.
4. List the raw and percentage results of a particular roll call vote by party division.
5. List the raw and percentage results of a particular roll call vote by region.
6. List the Index of Cohesion for each party for a single roll call vote.
7. List the Index of Cohesion for each region for a single roll call vote.

Students select one of seven modes of retrieval in response to the instruction: SELECT YOUR OPTION. One of the seven possible responses listed below which correspond to the seven modes given above, is typed in.

1. LBC
2. LSEN
3. LSTATE
4. %PARTY
5. %REG
6. CPARTY
7. CREG

An example of the LBC option is shown in Figure 1. RCVA also maintains a file for each student which records how many times he has used the system[8]. RCVA is currently being used to access the 21 roll call votes dealing with restrictive immigration legislation mentioned earlier as well as civil rights legislation in the 1950's and 1960's. Thus, there are two RCVA units for two different time periods.

The third computer assisted learning unit is VOTRAN, which is an abbreviation for Vote Analyzer. VOTRAN provides an on-line retrieval capability for the analysis of election returns and census data. The data files contain the information for New York, Michigan, and Nebraska described earlier in this paper. Students may select the state to be analyzed and then exercise a series of options which include:

1. List raw data for one or more counties on a particular variable.
2. Compute and print the percentage of a county's or a group of counties' total of the state total for a selected variable.
3. List the total of a variable for two or more counties on a selected variable.
4. List the state's total for any number of variables.
5. Rank any number of counties on a single variable and print out the rank order of all the counties or print out a selected number from the top, middle, and bottom counties.
6. Aggregate, divide, subtract, or multiply two or more variables.

The instructions for executing the options are:

1. Row or r
2. Percent or %
3. Ctotal or c
4. Stotal or s
5. Rank
find or f
list or l
top or t
middle or m
bottom or b
6. ?, +, /, -, or *

Figure 2 illustrates how a student used VOTRAN options 3, 4, 5, and 6 to access the New York data file. In this example the top five and bottom five counties on variable 71, the percentage Democratic vote for President in 1928, were retrieved through the rank option. These ten counties (the numeric code for each county is given in the learning activity package) were then ranked on variable 241, the percentage nonwhite in 1930, variable 247, the percentage urban

Select your option.
 LFC
 Which senator?
 num
 37
 Which roll calls?
 all

SENATOR	37
ROLL CALL	RESPONSE
1	YEA
2	YEA
3	YEA
4	ABSENT
5	NAY
6	YEA
7	NAY
8	YEA
9	NAY
10	YEA
11	NAY
12	NAY
13	ABSENT
14	ABSENT
15	YEA
16	YEA
17	ABSENT
18	ABSENT
19	ABSENT
20	YEA
21	ABSENT

FIGURE 1.

Select your option.

—
 rank
 Which variable?
 var

71
 Which counties?

1-62

One moment please.

What would you like?

top

How many?
 num

Rank	County	Data	Variable	71
1	3	68.300		
2	31	61.100		
3	24	59.500		
4	10	58.000		
5	1	55.000		

What would you like?

bottom
 How many?
 num

Rank	County	Data	Variable	71
58	9	22.000		
59	54	21.400		
60	13	21.100		
61	62	20.700		
62	2	18.000		

Select your option.

rank
 Which variable?
 var

241

Which counties?

3,31,24,10,1,9,54,13,62,2

FIGURE 2.

One moment please.

What would you like?

List

Rank	County	Data	Variable	241
1	31	12.600		
2	24	2.800		
3	1	1.300		
4	3	1.100		
5	10	1.000		
6	54	.600		
7	62	.600		
8	9	.500		
9	2	.500		
10	13	.300		

Select your option.

Rank

Which variable?

var

247

Which counties?

3,31,24,10,1,9,54,13,62,2

One moment please.

What would you like?

List

Rank	County	Data	Variable	247
1	3	99.800		
2	31	99.800		
3	24	99.800		
4	1	80.700		
5	54	40.600		
6	10	35.800		
7	62	31.600		
8	9	24.200		
9	2	14.500		
10	13	8.500		

What would you like?

Select your option.

rank

Which variable?

var

243

Which counties?

FIGURE 2. Continued

ANALYSIS OF PUBLIC OPINION

What is your identification number?

10

This is session 1 for you.

What survey do you want to investigate.

1964

In a moment you will compare the responses to any two questions selected from the list in your packet. After you specify which two questions are to be compared a frequency table for that pair of questions will be typed out. Then you will be asked if you want to see table percentages, column percentages, and row percentages. If you wish to see this information just answer yes or no to the questions. Remember that you can compare only two questions at a time.

Type the number of the first question under the label var1 and the number of the second question under the label var2.

What are the numbers of the two questions you want me to compare?

var1

2

var2

24

Table of Frequencies for Question 2 and Question 24

Row=Question 2

Column=Question 24

Row	1	2	3	4	Total
1	141	157	41	12	351
2	138	249	92	14	493
3	115	165	195	5	480
4	97	115	24	11	247
Total	491	686	352	42	1571

Do you want to see table percentages?

yes

Table Percentages for Question 2 and Question 24.

Row=Question 2

Column=Question 24

Row	1	2	3	4	Total
1	9.0	10.0	2.6	.8	22.3
2	8.8	15.8	5.9	.9	31.4
3	7.3	10.5	12.4	.3	30.6
4	6.2	7.3	1.5	.7	15.7
Total	31.3	43.7	22.4	2.7	100.0

FIGURE 3.

in 1930, and variable 243, the percentage foreign born stock in 1930. The object in ranking the counties on these variables was to determine if there was any significant shift in rank from that on variable 71, the percentage Democratic vote for President in 1928. The Ctotal or c option was then used to aggregate variable 65, the Democratic raw vote for President in 1928, for the top five counties on variable 71. Next, the Stotal option was used to retrieve the total state Democratic raw vote for President in 1928. The ? option then yielded the percentage the five county total was of the total state vote.

APO, or Analysis of Public Opinion, is the fourth computer assisted learning unit now being used to teach American history at the undergraduate level. Like the other units described previously APO is an on-line interactive system which permits students to access and manipulate data files composed of responses of a representative sample of the adult population interviewed by the Survey Research Center of the University of Michigan in 1960, 1964, 1968, and 1970[9]. Data file manipulation consists of comparing responses to a selected pair of questions and printing out a cross-tabulation frequency table, table percentages, row percentages, and column percentages.

The current APO data files include responses to between 45 and 55 questions selected from about 450 questions asked in each survey because they involve contemporary social and political problems of great interest. For example, several questions measure attitudes toward segregation and desegregation and civil rights activities of Blacks while other questions deal with American involvement in Vietnam.

Figure 3 displays the options available to students in APO. The 1968 survey was selected and a comparison of the responses to questions 1 and 55 was requested. Question 1 deals with the population of the place in which the respondent lived while question 55 asks how he voted for president. The table printed out lists how the respondents voted by population. Population areas are indicated by the row numbers at the left and vote for president is indicated by the column numbers at the top of the table. (The learning activity package for this unit explains what each row number and column number means.) Execution of the table percentage and column percentage options yielded the two other tables in Figure 3.

A more sophisticated version of APO is called TDAPO or Third Dimensional Analysis of Public Opinion. It uses the same data files as those in APO but compares responses to a pair of questions while controlling for a third question. Suppose, for example, a student wanted to know if southern women were more intolerant toward desegregation than were their counterparts in the North, the Midwest, or the West. The responses to questions of the region in which respondents lived and their attitudes toward desegregation would be compared while controlling for the question of sex. TDAPO would print out two cross-tabulation tables, one for female and one for male. Each table or plane would be broken down by region and attitude toward desegregation.

The sixth and last operational computer assisted learning unit is called Ghetto[10]. It is a computerized version of the educational game Ghetto which is a simulation of what life is like in the ghetto. It is designed to expose non-ghetto residents to some of the pressures at work in the inner city neighborhood. Each player selects one of ten roles of poor people and tries to improve that person's life. In this attempt he experiences vicariously some of the frustrations and deprivations that affect the lives of the urban poor.

Unlike the board game, the computerized version of Ghetto requires only one player. A player selects a profile and then describes himself to the computer in terms of responses to questions such as how old he is, how many children he has, what is his level of education, what is his sex, and how many hours he has to invest. (All of this information is given in the learning activity package of this unit.) After this description is typed in, a player then is asked how he wants to invest his hours. He may elect to invest his hours in school, work, hustling, recreation, and welfare. Since the object of the game is to score as many points as possible during ten rounds, a player normally makes an investment that brings the greatest number of points. Hustling, and especially big time hustling, offer great rewards but also run the risk of arrest. On the other hand, education offers great opportunities to improve future earning power and thereby earn more points.

Built into the computerized version of Ghetto are chance factors over which a player has no control. Illness, for example, is assigned by a random number generator. Pay cuts, being robbed, or being arrested if you hustle are also assigned on a random number basis. Each female runs a 16 percent chance of having a child, again on a random number basis.

A round is completed when an investment is made and the chance factors are generated. The computer totals the number of points earned. The sequence is repeated until ten rounds have been played and a total score is tabulated.

Thus far Ghetto has been played by pairs of players who select the same profile. The first player moves through the game following one strategy. The second player plays the game using another strategy. For example, the first player might concentrate upon hustling while the second

player focuses upon work or education. At the conclusion of the game they could compare results. This tends to reinforce the lesson that while education usually is a vehicle for upward mobility, it is exceedingly difficult to do this in the inner city. Hustling generally provides the largest reward.

Each of these units is operational and has been tested in one or more of the following history courses at Oklahoma State University: History 4503, The American City Since 1865, is a junior-senior level course; History 2493, American History Since 1865, is a freshman-sophomore level course; and History 4183, American History Since 1920, is a junior-senior level course. Student participation in testing these units was voluntary, although in the two upper division courses students were strongly encouraged to participate. In the Spring semester of 1972, students enrolled in History 4183 will be required to complete these six units plus two others now being developed. Student participation in History 2493 in the Spring semester will continue to be on a voluntary basis.

Section III

At this point in the paper attention can be directed to two major questions: (1) What was the cost of using these six units? (2) How did students respond to computer assisted learning? The first question can be readily answered and with considerable accuracy. Table I summarizes what is called developmental costs. Included in the table is a unit called IVIS which was dropped after having been developed to the point of being operational. Excluding this unit, the cost of programming, CPU, and data file creation was \$2,335.00. Almost one-half of this (\$1,000) is accounted for by creation of the APO files of 1960-1970. There are several reasons for this, one of which is that considerable recoding of the survey data is necessary. More important, however, is that each file contains cross-tabulations of the responses to each question with every other question (there are 1486 such cross-tabulations) with answer codes that range from 1 to 10. Of course this is a one time expenditure.

<u>Unit</u>	<u>Programming[a]</u>	<u>CPU</u>	<u>Data File Creation</u>
RCVAI	\$125.00	\$ 25.00	---
RCVAII	25.00	10.00	---
Votran	150.00	50.00	\$ 200.00
IVIS[b]	145.00	200.00	---
APO	225.00	150.00	1000.00[c]
TDAP0	100.00	50.00	125.00
Ghetto	75.00	25.00	---
Totals	\$700.00	\$310.00	\$1325.00

[a] not included in totals

[b] programming charge is \$5.00 per hour

[c] this includes five separate files which initially are created in the batch processing system at a cost of \$200.00 per file

Table II displays a report of approximate instructional costs of using four of the six units in an actual teaching situation. Inspection of the table discloses that with the exception of Ghetto, Votran, APO, and TDAP0 are quite costly, although this can be reduced by increasing the number of users. The TDAP0 unit costs more per student simply because of the extensive computer sorting that is necessary in controlling for a third variable. On the average about two minutes of CPU (\$100 per hour) are required to generate one comparison. This in turn translates into about fifteen minutes of elapsed terminal time (\$1.80 per hour). Notice that there is only a difference of \$1.79 per student between lower division and upper division users. This is not as much as one might expect, given the fact that there were four more users in History 4503. A comparison of the mean CPU and mean terminal time for the two groups discloses that students in History 4503 used on the average almost one minute more of CPU during almost the same average terminal time. This suggests that junior and senior students made much more efficient use of terminal time. Also, a comparison of the hypotheses the two groups tested indicates that on the whole students in History 4503 were much more complex.

The APO unit instructional costs vary considerably. The cost per student user in History 4503 was \$8.54 during the summer and 12.79 in the fall. The lower cost is explained by the fact that only two survey data files were available during the summer and that students during the fall averaged one minute and twenty seconds more of CPU and twenty minutes more of terminal

time. This suggests that students in History 4503 during the Fall semester posed hypotheses that required more elaborate data retrieval. When the cost per student figure for History 4503 and History 2493 is examined, the former is almost twice as high. The larger number of users (22 as opposed to 10) in History 2493 helps to account for this along with the fact that upper division students tended to make more effective use of terminal time. Despite this, the cost per student is still quite high. The explanation for this is the five APO data files are enormous, occupying in excess of 550 disk tracks at a cost of \$5.50 per day for data and system storage.

Votran is a rather expensive unit also. Interestingly enough, the major cost of Votran is CPU time. There are two reasons for this. First, twice as many students used Votran as used APO or TDAPO. Second, the rank option in this unit simply requires extensive sorting which uses considerable CPU time. It might be noted also that after the rank option is executed, about five minutes of terminal time (this varies with the number of terminal users) elapse before any information is typed out.

A review of the computer aspects of Votran, APO, and TDAPO indicates that the cost per student can be reduced somewhat. If the one-line rank option of Votran is eliminated, this would likely cut the cost in half. In order to do this a new data file will be created in which every variable is stored both in rank order and in natural order. This will increase the size of the data file by about one-third with increased storage costs, but this will be more than offset by sharply reduced CPU time. The cost of APO can be lowered by shortening the amount of time that the system is up. Probably a limiting access to APO to twenty-one days (the information in Table II is based on thirty days) students will encounter no difficulty. However, a period of time less than this might create hardships for some students. TDAPO cost per student can be reduced by having students use the remote job entry (RJE) feature of CPS to enter a job into the batch processing stream. The print out may be picked up several hours later at the Computer Center. There is some question as to whether reduction in cost will not offset the inconvenience of waiting several hours to analyze the data.

Reference was made earlier to IVIS, a unit that was dropped. From a programming point of view IVIS was quite elegant, chiefly because of its remote job entry (RJE) capability which permitted students to enter a job in the batch stream from a remote terminal with three simple executions. Despite its elegance, however, IVIS was dropped because test runs indicated that the kind of jobs processed in this way would cost in excess of \$30.00 for each job. There was no way to justify this in a class of thirty students. Therefore, IVIS was relegated to the junkpile of impractical projects. It was an expensive lesson, but one well learned. It might be noted in passing that IVIS was not a total loss. The remote job entry (RJE) of IVIS can be added to TDAPO with a few modifications, although it too may prove impractical.

Earlier in this section of the paper two questions were posed, one of which dealt with costs and the other with how students responded to the computer assisted learning units. The cost factors have been discussed. Attention may now focus upon the responses of nineteen students in History 4503 to a survey of their attitudes toward computer assisted learning. The following statements summarize the results of this elementary survey.

1. Only one student indicated that computer assisted learning was not useful.
2. Interestingly, ten students considered the four computer assisted learning units less difficult than conventional textbook assignments of comparable duration. One student noted that the units were less difficult because "you can see what you're doing."
3. In terms of preference for the four computer assisted learning units one student had no preference (this was the same student who considered computer assisted learning not useful), three students preferred Votran, five favored APO, five chose TDAPO, and six selected Ghetto.
4. Only two of the students indicated they were not uncomfortable and ill at ease when they first used the terminal. Both of these students had had previous experience in using a remote terminal.
5. Fifteen of the students reported that by completion of the second unit they were no longer uncomfortable or ill at ease.
6. When asked if they could choose between enrolling in a class taught without computer assisted learning units and a class taught with these units, fifteen said they would enroll in the latter. Several students added the comment that a course utilizing computer assisted learning units would be more interesting and enjoyable.

Additional student feedback disclosed that the two class sessions devoted to briefing students to the units and how the remote terminal works were insufficient. It was the consensus of the group that they felt like a non-swimmer who had been taught to swim by being thrown into a body of water and told to swim. More attention must be given to this problem as well as to urging students to be less impatient while awaiting completion of a rank option or a TDAPO execution.

Apparently, a number of students were bored by this aspect of a five or ten minute delay in the information being typed out.

An appropriate note on which to conclude this paper is that of observing that this paper reports findings that only point toward the kind of accountability required. More information on costs and student attitudes is essential. In addition, differences that computer assisted learning makes in what students learn and how they learn must be documented. The project at Oklahoma State University will attempt to provide some of this information and documentation. In the Spring semester of 1972 a total of eight computer assisted learning units will be used in History 4183 and History 2493. Approximately 75 students will be involved. Also students in History 4183 will participate in an evaluation of the effectiveness of computer assisted learning in enhancing critical thinking abilities. The hypothesis to be tested is that students exposed to computer assisted learning as described in this paper will register improvement in critical thinking abilities. A pretest and posttest using the Watson-Glazier Test of Critical Thinking will measure this improvement. Obviously, this will be a very crude measure since there are too many uncontrolled variables which might intervene. However, in both the Fall of 1972 and the Spring of 1973, one class in American history will be taught using conventional classroom instructional approaches and another class (the same course) will use the computer assisted learning units described in this paper. It is anticipated that by using an experimental class and a control class the results will have greater validity.

FOOTNOTES

1. Proceedings, Conference on Computers in the Undergraduate Curricula, 1970 (University of Iowa), pp. 0.1-0.3.
2. Proceedings, Conference on Computers in the Undergraduate Curricula, 1971 (Dartmouth College).
3. For example, the National Science Foundation is supporting EDUCOM's "Investigation of Factors That Inhibit Widespread Use of Computers in the Instructional Process."
4. Edwin Fenton, The New Social Studies (New York, 1967), pp. 6-27.
5. This information in machine readable form was obtained from the Historical Archives of the Inter-University Consortium for Political Research, University of Michigan, Ann Arbor, Michigan.
6. CPS stands for IBM's Conversational Programming System and is a time-sharing mode of computer-user interaction.
7. These two units have been tested and are operational. However, they are not included in cost analysis later because their substantive content deals with a course taught in the Spring of 1971.
8. Certain constraints of CPS make it difficult to obtain a record of the actual amount of CPU and terminal time used. Students turn in the last page of their typed output which includes a record of CPU and terminal time. Sometimes, students lose this piece of paper or forget to turn it in. This particular feature of recording how many times each student uses the system (each student has a separate identification number) will disclose any discrepancy between actual student performance and the information he turns in.
9. This information in machine readable format was obtained from the Survey Research Archives of the Inter-University Consortium for Political Research, University of Michigan, Ann Arbor, Michigan.
10. I am grateful to Academic Games Associates, Inc. for granting me permission to develop a computerized version of Ghetto.

EDUCATIONAL USES OF
PRINCE

William D. Coplin
Michael K. O'Leary
Syracuse University
Syracuse, New York 13210

PRINCE, a programmed international relations computer environment, operates as a man-computer simulation in which the student plays the United States foreign policy-maker and the computer plays five other states as well as domestic political pressures within the United States. It also operates as an all-computer simulation modeling international political processes. PRINCE serves as an educational tool with a wide variety of applications. The categories of uses discussed below are not mutually exclusive. Two or three educational applications might be pursued simultaneously which may, in fact, be the most desirable way to proceed in many educational settings. We hope, however, to suggest a list of the full range of educational uses that might be made of PRINCE as it currently operates.

As an educational tool, PRINCE can help achieve five general goals:

1. Introducing the student to basic facts about contemporary international relations;
2. Providing the student with a framework for understanding those facts;
3. Introducing social science methods and informing the student of current theory and research in international relations;
4. Improving the student's decision-making and planning skills;
5. Helping the student to deal with the problems of effectively organizing a group to undertake complex decision-making.

Each of these uses will be briefly described below.

Introduction to the Facts of Contemporary International Relations

At its least complicated level, PRINCE can be used as a vehicle to get the student to learn and digest basic facts of the contemporary world. The model is regularly kept up to date so that any play starts with the present international situation and extends into the immediate future. A student who plays the model in September, for example, will be asked to make policy regarding the issues which are current at that time. The student can be asked to study newspapers and other periodicals to determine the state of those issues being dealt with as preparation for making appropriate decisions in the simulation. A perfunctory reading of the New York Times will not suffice. The student will be encouraged - and, on the basis of our experience, actually motivated - to seek out information with as much, if not more, enthusiasm as the Foreign Service Officer does every morning of his life.

This use of PRINCE is the most direct and least complicated of the six uses. It can be successfully achieved with as few as three cycles of the PRINCE model and two weeks of a regular three-hour course. At Syracuse University, the Political Science Department has prepared a one credit "mini-course," entitled "The United States in World Affairs," which is open to all undergraduates. The students are divided into teams of four. Each team is asked to brief itself on a particular set of issues and countries in the model and to play six cycles. Resources available to help the students prepare themselves for play include tape-slide presentations describing the foreign and domestic politics of states, extensive newspaper clipping files, and a bibliography. After six cycles, the students submit a report on their policies and their analysis of the results of their "tenure in office."

In this particular educational mode, PRINCE is being used primarily to provide an intriguing environment within which students will systematically study substantive facts about contemporary international relations. Such use would be most appropriate in general college courses and perhaps even high school courses dealing with either United States Foreign policy or International Relations. It can be organized as a short course on its own, or included in a full-term course. Because its educational objectives are focused on the acquisition of basic information, this use of PRINCE might also be appropriate for high school courses as well as adult education.

A Framework for Understanding International Relations

In addition to providing a context to motivate students to acquire the facts of international relations, PRINCE can also be used as a means for helping the student acquire a basic conceptual framework. In this case, the educational aim is not to help the student acquire information about contemporary international issues, but rather to provide him with a set of concepts that will help him make sense out of the information available to him. Basic concepts like international bargaining, diplomatic pressure, transaction flows, and domestic political influencers are introduced to the participant as he attempts to deal with the simulated international environment. After playing the exercise a number of cycles, the student should be able to describe international relations in terms of these and other related basic concepts.

This use of PRINCE requires only a few cycles of play followed by a thorough discussion in which the basic concepts are related to the writings of international relations scholars and commentators and to the events as reported by scholars and journalists. The student might be asked to describe the Middle East Problem in terms of sets of issue positions among the states and domestic political influences within states, or he might be required to relate the policy suggestions of a writer like Henry Kissinger or Thomas Schelling to the PRINCE world. The use of PRINCE is typically employed in introductory courses on international relations or foreign policy. It provides an alternative to the normal procedure of providing a theoretical framework through textbooks or lectures. It can, of course, be used to supplement and compare concepts which are supplied in the traditional fashion.

An Introduction to Social Science Method and Literature in the Study of International Relations

A third and more complex use of PRINCE is to provide the student with an introduction to the methodological and theoretical problems involved in the social scientific study of international relations. After learning the basic PRINCE concepts through playing a few cycles of the simulation, the student is then introduced to the concepts and theory of which the computer model is composed. In addition to the concepts visible to the player (e.g., issue position, influence attempts, transactions and domestic political influencers), there are other concepts such as affect, salience and power that form the theoretical infrastructure of the model. Additional ideas such as dependence, the irresponsibility ratio, and the reference ratio, which are derived from the basic concepts of issue position, power, salience and affect are also described. The student is provided with both verbal and written descriptions of the concepts and theories in the model.

In the context of this description, the student is then provided with the opportunity to work on a variety of problems. First, and at the most general level, he can see the utility as well as the limitations of models in the study of international relations. Second, he becomes aware of the challenge that must be faced in testing the validity of the theoretical ideas in the model. The tasks involved include collecting data from appropriate sources, selecting correct measurement techniques, and relating existing theoretical and research concepts to the operations in the model, and the choice of appropriate statistical techniques for testing the relationships between variables. Third, he can start to test some of those ideas himself and suggest alterations in the basic model based on the ideas of other scholars as well as his own thinking. From the perspective of the PRINCE model, the student can be introduced to advanced and sophisticated approaches to the social scientific study of international relations.

Our experience suggests that this use of PRINCE is best reserved for more advanced students who have had some previous work in international relations. Juniors, Seniors and beginning graduate students have been introduced to the research and theoretical literature on international relations through this approach. They have critiqued particular aspects of the model for failing to conform to the theories of scholars and have tested the validity of certain theoretical relationships using data from a variety of sources. Given the complexity of the task, the model can be utilized in this mode for a substantial period of class time. In the past, we have found that up to eight or ten weeks of a normal three-credit advanced course can profitably be used in these tasks. When supporting analytical materials are developed, an entire three-credit course will be built around these activities.

Providing the Student with the Opportunity to Improve His Decision-Making and Planning Skills

This use of PRINCE shifts from providing information, a framework or an approach for understanding other scholar's framework to helping the participant to be able to handle better decision-making under complex and uncertain conditions. PRINCE represents a highly complex environment in which the player must learn to deal with a large amount of information which often appears at first to have no discernible pattern. He also is confronted with an environment where multiple and often competing goals are suggested.

The player must learn to deal with the five classic tasks of decision-making: (1) defining the situation, (2) selecting goals, (3) examining alternatives, (4) choosing alternatives and (5) determining the consequences of decisions. Evaluating the skill with which these tasks are

performed can best be done in the context of a dynamic and complex simulated environment which can be repeated indefinitely. The player learns the necessity of developing explicit models about the world that can be continuously checked and readjusted. He sees the value in making his goals explicit so that he can determine what information is important to him and what is not. He can explore a variety of strategies so that he can learn to project the consequences of alternatives. And finally, he is given practice in making choices in a complex environment where only the bare outlines of the probabilities can be ascertained.

This use of PRINCE can be undertaken in a variety of educational contexts. Most students are eager to try their hand at decision-making; PRINCE can provide him with a surrogate environment which is immediately challenging and yet complex enough to warrant more thoughtful analysis. Similarly, mid-career training in a number of fields has come increasingly to emphasize the development of decision-making and planning skills. The PRINCE environment provides the kind of tool that the mid-career professional can use in learning to make complex decisions with uncertain information. Like the college student, he can be shown the benefit of developing intellectual guesses formalized in tentative models to deal with decision and planning problems. At the college, graduate or mid-career levels, however, the student should be allowed at least eight cycles of play so that he can learn to deal with sequential and fairly long-range consequences of policy decisions. He should also be taught to develop and apply criteria for judging the efficacy of such policies through a series of mapping and analytical exercises.

Problems of Organization in Dealing with Complex Decision-Making and Planning

A special problem which should not be overlooked in group decision-making and planning is the need for specialization, divisions of labor, and communications. How should a group organize itself to deal with a complex environment? This question is critical to the success of either understanding or dealing with contemporary political and social decision-making. PRINCE can be used as a source of a complex decisional task around which group structures can be built. Five to ten individuals can be organized with geographical and/or functional responsibilities to deal with the PRINCE environment. In addition, roles such as information gathering, synthesis, policy advocacy and even evaluation can be developed and introduced as part of the group structure.

This is a particularly appropriate use in the study of the politics of decision-making in large-scale organizations. It fits well into advanced undergraduate and graduate courses as well as in mid-career education. It exposes the participant to the problems of coordinating and controlling members of a bureaucracy as they seek to deal with a complex environment. We recommend that at least ten cycles be undertaken since a relatively substantial history of relationships must develop in order to have the organizational factors take effect. Again, a substantial part of a course or a two or three week period of intensive work is required. At the end of that time, the writings of organizational theories can be tested by the experience of the group as well as skill and knowledge in handling the roles of decision-making in complex organizations.

SUMMARY

We have briefly addressed the various uses of PRINCE in a variety of educational settings. As we have already noted, it should be clear that more than one use can be made of PRINCE at one time. It should also be clear that different educational uses should be applied to different types of students. The college freshman would not benefit from learning about decision-making in complex organizations via PRINCE while the mid-career professional does not need PRINCE to help motivate him to read the newspaper. Nevertheless, the educational objectives of instructors at virtually every level of sophistication can employ PRINCE in a useful manner.

To help carry out these objectives we have begun the development of a wide range of multimedia educational materials. Central to all the objectives mentioned is the Participants' Guide to PRINCE: Concepts, Environments, Procedures which is available for distribution. It is periodically brought up to date as issues change in the international environment. The Guide outlines the basic domestic and international environment with which the player must deal, details the alternative policy actions available to him, and describes the basic procedures to be followed in operating the simulation.

Video tapes and newspaper files have also been prepared to provide "country briefings" for the student who is playing PRINCE within the context of acquiring basic information about contemporary international affairs. A series of analysis forms is being field tested at this time by which the student can record and study the history of his own policies and the consequences of his actions. The tapes, files and forms will be available for distribution in the fall of 1972.

Materials are also being developed to aid the student in understanding the theoretical structure of the PRINCE model. A draft version of the description of the model is now available ("A Brief Description of the PRINCE Model") and a final revision will be available in June, 1972. To supplement this description, William D. Coplin and Michael K. O'Leary will publish in the Spring of 1972 a book entitled "Everyman's PRINCE: A Guide to Understanding Your Political Problems" (Belmont, California; Duxbury Press, A Subsidiary of Wadsworth Publishing Company). This book introduces the basic PRINCE concepts in the context of familiar domestic and international political situations. In addition to the Description and Everyman's PRINCE, a Workbook is now being developed that relates empirical data and historical case studies to the various concepts and theories in PRINCE. This workbook will be in draft form by September, 1972 and will provide sufficient material so that an entire three-credit course can be provided to the student as independent study.

The following materials related to PRINCE are available as of January 1, 1972 from the International Relations Program, Syracuse University, 752 Comstock Avenue, Syracuse, New York 13210:

"The PRINCE Project and the International Relations Program," this short paper outlines the basic research and educational goals that constitute the activities of the International Relations Program and PRINCE.

PRINCE Participant's Guide: Concepts, Environments, Procedures by William D. Coplin, Stephen L. Mills and Michael K. O'Leary. Single orders sent upon request; Bulk Orders 75 cents each.

"A Description of the PRINCE Model" (\$1.50 each)

Everyman's PRINCE: A Guide to Understanding Your Political Problems (Obtain from Wadsworth Publishing Company, Belmont, California after April, 1972).

VOTER:

A SOCIAL SCIENCE DATA ANALYSIS PROGRAM

Bruce D. Bowen and Wayne K. Davis
The University of Michigan
Ann Arbor, Michigan 48104
Telephone: (313) 764-0327

Introduction

VOTER is an example of an instructional use of the computer which accomplishes educational objectives probably not possible by any other means. The VOTER program was written for use at a remote terminal for the unsophisticated student user to develop his ability to analyze and interpret large sets of data. The program has been specifically designed to be used in undergraduate political science courses. This paper will describe two different uses of VOTER in these courses.

VOTER has been designed to achieve the following instructional objectives:

1. to enhance the student's understanding of the theory, practice and use of survey statistics through direct involvement with real data bases;
2. to encourage the development of those intellectual abilities required by the process of uncovering and interpreting information which is effectively new;
3. to provide direct contact with a computer, at a level which is appropriate given the student's background and needs, in a worthwhile enrichment experience.

VOTER can be used by the instructor as a stimulus for discussion, a device to motivate and facilitate individual study, a source of factual content for course work, or a research tool for project-oriented assignments on an individual or group basis. The cost of using the program is minimal; its educational value to the students are largely a function of how it is used by the individual instructor and student.

Description of the Program

VOTER is designed to generate labeled, univariate and bivariate frequency tables selected by the student. VOTER is data base-independent since practically any data base can be prepared as input, and routines have been written to prepare new data sets for analysis by the VOTER program. The program provides for selecting samples and subsets of the data base that has been entered. In addition to producing univariate and bivariate frequency tables, VOTER also computes correlational statistics to accompany tables selected by the student.

At different points in the program the student can select samples or subsets and determine the type of table or statistic to be computed. One way to describe the operation of the program is to explain the options available to the student at each point in the program's execution. An attempt will be made to describe these options as the typical student works his way through the program.

After initiating the VOTER program, the first option open to the student is the selection of a sample of the data with which he wishes to work. The option consists either of the entire sample or some subset of the sample. If the student selects the entire sample he is given another option. This option consists of naming a column variable. If the student selects a subset of the entire sample, the selection of a column variable is postponed until the subset is identified. Naming the column variable is accomplished by entering a variable name which not only selects the column variable but also retrieves the stored labels that will be printed with the table. After selecting the column variable by name, the student has the option either of entering the name of the row variable for a bivariate table or entering "none" to indicate that he would like a univariate table.

If the student enters a row variable he is next asked to determine the independent variable. The selection is important for two reasons. First of all, the percentages that are printed with the frequencies in the table are given in terms of the independent variable. If for example, the student indicated that the row variable was the independent variable the percents would be given so that each row of percentages would sum to 100. If the student identified the column variable to be the independent variable, each column of percentages would sum to 100. The student may also select "total" which would cause the percentages to be computed so that the sum of all percentages in the table would be 100. Secondly, naming the independent variable insures the calculation of the proper asymmetric statistics. After the table has been printed by the computer, the correlational statistics are given to help the

student in the interpretation of the table. These statistics are Kendall's tau, Gamma, Somer's D, and Goodman and Kruskal's Lambda.

After the statistics for the table have been printed, the student is allowed to make another choice. This choice is either to (a) start again with the entire sample, or (b) continue generating tables with the existing sample or subset as he has already described it, or (c) make additions or deletions to the current sample (or subset) using the filter procedure. Also at this point or any other option point in the program the student may enter "stop" and the program will terminate. If the student would like to do filtering, he would enter the response "more" indicating that he wishes to enter a filter command.

The filter command consists of three pieces of information: the variable name, either the keyword "include" or "reject", and the response code number. If he types "reject", all cases with the code number specified will be deleted from the sample. If he types "include," all cases with the specified code number will be included in the sample along with any other cases that have not been explicitly deleted.

The student is now faced with an option he has seen before, the selection of the column variable. This is once again done by typing the name of the variable to be represented in the columns of the finished table. The student is then asked to enter the name of the row variable. He may enter either the name of the row variable or "none" indicating that a univariate table is desired. If he has indicated a row variable he will then be asked to specify the independent variable. After this table has been printed, the student is allowed either to select a fresh sample, to filter further, or to obtain another table from the same sample.

The program contains a special error checking routine which scans every student response for possible errors. When an error is detected the message "Your response was unrecognizable. Please try again." is printed, and the program is made ready to accept a new response.

VOTER in Context

The program has been used in two different course settings. In both the fall of 1970 and 1971 the program was used in a junior-senior level course entitled "American Political Process" in the University of Michigan Political Science Department. In addition, in fall 1971 it was used in another junior-senior course, "Public Opinion and Pressure Groups." Less than 18% of the students enrolled in these classes reported any previous experience with a computer.

The first course is a large survey course in which the students are almost evenly divided between political science majors and non-majors. While the non-majors are frequently from one of the other social sciences, the course does fulfill the college distribution requirement for the social sciences, and therefore attracts students from many areas. Due to the large enrollment (100-150 students) this course is typically taught using a lecture format. The content of the course concentrates on mass actors (as opposed to elites) in the political system. Some of the major topics studied are voting behavior, public opinion, and political parties.

In contrast the course "Public Opinion and Pressure Groups" is composed of mainly political science majors (83%). Its focus is upon the measurement of public opinion, formation of opinions, political socialization, models of opinion change, demographic bases of opinion, and psychological models of opinion change. This course is normally smaller than the political process course (20-50 students) and is typically characterized by discussions and more student input.

Because of the differing foci of the two courses, the computer program was used differently in each course. In the course on the political process which focuses on voting behavior, the students were assigned data based term papers. The VOTER program was used to analyze one of the current election studies of the Center for Political Studies (C.P.S.) (1). These analyses formed the principal components of the term papers. The students were relatively free to select a topic within the range of material covered by the base study. During fall 1971 students were using data sets composed of 100 variables from the CPS 1970 Election Study.

By using the cross tabulation features of the program (bivariate frequency table) students were able to investigate the relationships between several pairs of variables, while the filter option allowed students to control additional variables. The combination of these two features enabled students to conduct a wide variety of analyses at a relatively sophisticated level.

The public opinion course constructed a data set based upon interviews of Ann Arbor voters collected by the students themselves. In this way the students were able to experience first-hand survey research through drawing a sample, conducting interviews, coding the questionnaires, and analyzing the results by means of this program. As in the political process class, the students were asked to write term papers based on their analyses of the data. In addition to

the final interview results, the public opinion students were asked to analyze a data set from the pretest of their survey instrument.

Because of their greater involvement in the data gathering and data analysis phase, the students on the public opinion course were even more enthusiastic about the program than the students in the political process course. Some students became so interested in the analysis of their data that they requested additional assistance from the instructor to conduct some analyses that were too sophisticated for the VOTER program. This additional involvement was much easier for this class because it was a smaller class composed mainly of political science majors.

In both classes initial instruction in the use of the computer program and the Michigan Terminal System (MTS) were provided by the Department of Political Science Computer Assistance Program. Throughout the semester both the instructor and the graduate students in the Computer Assistance Program provided counseling in both the program and MTS. This was designed to provide the students with assistance when they needed it. The computer assistants were available Monday through Friday, 9 a.m. to 5 p.m. Before actually using the machine many students seemed apprehensive. Informal student reactions to the use of the computer seemed quite good, however, after an initial success at the terminal.

In order to gauge the value of the computer experience a survey was conducted of the students in the two courses offered in the 1971 fall term. The results of that survey confirmed that the use of the program had been quite successful. When asked of their general evaluation of the computer experience, 56% rated it "very favorable", on a four-point scale running from (1) "very favorable" to (4) "very unfavorable." The mean score was 1.45.

The ratings of the program and the general experience did not vary between classes and did not vary between those who had had some previous experience with the computer and those who did not. There was no variation between political science majors and non-majors. All groups preferred the data-based term paper to the more traditional library term paper by a substantial margin. (85% preferred the data paper.) The students felt the data paper was easier, more enjoyable, and more valuable. 62% thought the data paper easier, 90% thought it more enjoyable and 88% thought it more valuable.

The general evaluation of the computer program and its documentation was also quite high. When asked to rate how difficult it was to learn to use the program on a four-point scale from (1) "very hard" to (4) "very easy", the mean score was 3.17. The mean for those with no computer experience was also quite high, 3.10.

One of the surprising findings of the survey was that 68% wanted to find out more about the computer itself. Even a substantial percentage of those who had previous experience with the computer (58%) were not yet satisfied and wanted to learn more.

Limitations

One of the most significant limitations on the program is that a variable may have a maximum of five code categories. The reason for this is that the width of the output must be limited to 70 characters in order to fit on the Model 33 Teletype in general use at the University of Michigan. In an environment capable of handling longer line lengths, this problem could be easily avoided.

Some student frustration was generated by the inability of the VOTER program to recognize keywords and variable names entered with minor misspellings. Although most current programs suffer similar student criticisms, recent software developments could be utilized to allow for minor typographical errors and misspellings. Routines which allow for these errors are currently being developed for future use in VOTER.

Other Uses of the VOTER Program

The VOTER program should be used in a course where analyzing real data makes sense and where the teacher's objectives include increasing student competency in analyzing and interpreting data. Courses in the social sciences are obvious places to use VOTER; educational research courses, sociology, educational technology, research and design courses, political science, survey research courses and possibly even courses in applications programming are also suitable.

An important aspect of VOTER is that it was designed as a tool to improve instruction and not as a substitute for the textbook or the teacher. Therefore, it can be used in a variety of courses and is not subject matter dependent nor is it dependent on a particular textbook, or a particular view of teaching. It is an adjunct use of the computer in teaching and as such has a broad base of possible applications.

ACKNOWLEDGEMENTS

The VOTER program was written by Mr. Mark Barnett of the Center for Research on Learning and Teaching at the University of Michigan (currently at Ball State University). This work was supported in part by a grant from the Air Force Office of Scientific Research (AFOSR-68-1601).

We would like to thank Herbert Weisburg and Ronald Inglehart for allowing us to conduct VOTER evaluation surveys in their classes.

FOOTNOTE

1. Some of the data used by the course on the political process was provided by the Inter-University Consortium for Political Research at the Institute for Social Research of the University of Michigan.

A STUDENT HANDBOOK OF EMPIRICAL EVIDENCE: THE UTILIZATION OF
CAPE DATA IN UNDERGRADUATE EDUCATION

James E. Harf
The Ohio State University
Columbus, Ohio

Preface to the Instructor

This student handbook and data set were created in keeping with one of the five principal missions of the Project for the Comparative Analysis of Policy Environments (CAPE), under the supervision of Professor Philip M. Burgess, Director of the Ohio State University's Behavioral Sciences Laboratory:

...the utilization of the research environment in undergraduate teaching by creating an accessible data-rich setting with appropriate tutorial assistance to facilitate the empirical work of the undergraduate student...

It is expected that this student handbook will be utilized in conjunction with an international relations laboratory manual for undergraduates, Theory, Data, and Analysis: An Introduction to Quantitative International Politics (Burgess and Peterson, Allyn and Bacon, 1972).

The Burgess-Peterson manual introduces the student to a variety of ideas and concepts about how international data can be generated and analyzed in order to help answer some of the basic questions confronting scholars of international politics. Specific topics include basic considerations in the philosophy of science, principal data-generating techniques, elementary data-analysis techniques, fundamental considerations of research design, and instruction in the operation of an IBM Key Punch and Counter-sorter.

The student manual attempts further to advance a major goal of both the CAPE Project and the international politics faculty in the Department of Political Science: to develop a strong interface between faculty programmatic research endeavors and graduate/undergraduate education and training. Most of the international politics faculty have been engaged in the comparative study of foreign policy. Specifically, research has focused upon three principal themes:

1. The exploration of the theoretical and empirical utility of a typological strategy for understanding and organizing knowledge about nation-states.
2. The empirical utility and theoretical extension of one particular typological framework, Rosenau's pre-theory of foreign policy.
3. The development of satisfactory conceptions and empirically useful measurements of foreign policy outputs.

Consequently, a student data set has been created from a larger data file, the CAPE Project data base, which consists of a data matrix that reads 116 x 150. The 116 nations comprising the sample size are those national political units that were members of the United Nations in 1963 or that were excluded from U. N. membership because of their divided status. The approximately 150 variables in the data set represent principally aggregate social accounting indicators and external behavior attributes from the time-span of the 1960's.

From this ever-expanding list of CAPE variables, 72 have been included in the student handbook. This particular set of variables was selected because each appeared to tap specifically one of the Rosenau variable clusters in his pre-theoretical framework of foreign policy. The list was limited to 72, each expressed in one-column fields, in order that the data set would contain only one IBM card per country.

The number of variables for each variable cluster is given below.

<u>Cluster Name</u>	<u>Number of Variables</u>
Size	4
Development	4
Political Accountability	3
Governmental	7
Societal	7
External	10
Political Outputs	17

Economic Outputs	4
Military Outputs	4
WEIS Outputs	<u>12</u>
Total	72

The complete list of variables, their definitions and sources, and the coding scheme are given in the manual.

Eight of the 72 variables were originally in ordinal form. The remaining 64 were expressed at the interval level of measurement initially, but were transformed to the ordinal level by the following procedure. All interval variables were normalized by appropriate data transformation techniques, then standardized so that their mean equaled 100 and a standard deviation equaled 10. The range was compared for each standardized variable, then divided into 10 equal intervals (deciles). Ordinal value (from 0 to 9) were assigned to each decile. The values for each country on every variable were placed in the appropriate decile, and the decile number was assigned as the ordinal code for each datum.

Once the student becomes familiar with the data set, he can perform any type of data-analysis requiring either nominal or ordinal level of measurement. The Burgess-Peterson laboratory manual suggests a variety of nominal and ordinal associational statistics. Additionally, it can be argued that because the assumptions required for interval-level tests, such as regression analysis, have been met by the techniques employed to convert these data to ordinal measures, data analysis techniques normally associated with interval data may also be performed.

The variables in this data set have been assigned to columns 1-72. Columns 73 and 77 are blank. Three-letter and three-digit IRA codes for each country are given in columns 74-76 and 78-80 respectively.

The data set punched on IBM cards may be obtained from the Behavioral Sciences Laboratory.

Part I: Introduction to the Student

This handbook is designed to help the undergraduate student engage in his own research endeavor in order to uncover information about the how's and why's of a nation's behavior in the international arena. It is assumed that this manual will be utilized in conjunction with a companion volume: Theory, Data, and Analysis: An Introduction to Quantitative International Politics, (Philip M. Burgess and Lawrence E. Peterson, Allyn and Bacon, 1972). Whereas, the Burgess-Peterson manual describes how one executes a research design (theory-building, data-generating, and data-analysis techniques, etc.), this handbook describes a specific framework for understanding foreign policy behavior, and generates and explains the use of a data set created from a larger data file in the Ohio State University's Behavioral Sciences Laboratory specifically for undergraduates. By using both this and the Burgess-Peterson manual, you should develop some preliminary skills for doing your own research, as well as appreciate the many problems associated with such endeavors.

Part II: Nation and Variable Classification Schemes

If we are going to search out the influencers of nation behavior, or at least ascertain which set of conditions is present when a certain behavior pattern occurs, we need a framework which will allow us to distinguish among various types of nations. In this fashion we can investigate the influencers of behavior for a selected subset of nations as well as for the entire universe of nations.

This framework should also allow us to make some initial judgments about which kinds of influencers are important. That is, the framework should identify variables or clusters of variables which ought to be utilized in any analysis of foreign policy behavior.

One researcher at Ohio State, Professor Rosenau, has suggested that nations ought to be classified or typed according to three dichotomized attributes (what he terms "genotypic variables"): (1) the size of a nation (large and small); (2) its level of development (developed and underdeveloped); and (3) political accountability (open and closed societies). If these three national attributes are combined, eight types ("genotypes") of nations emerge. These are given in Figure 1.

Large				Small			
Developed		Underdeveloped		Developed		Underdeveloped	
Open	Closed	Open	Closed	Open	Closed	Open	Closed

FIGURE 1. Nation Genotypes

A variable strategy which you may wish to adopt in your research is to use as your sample size only those countries falling into one category of nations. For example, you may wish to utilize only one attribute such as size in order to examine the behavior of nations within either the large or the small group. Alternatively, you may use all three genotypes to create a subset of nations, such as the large-developed-closed group of nations, for the purpose of analyzing behavior. Consequently, Figure 2 reveals the genotypic category for each nation in the year 1963. The specific measures utilized for each genotype are given below:

- size - total population
- development - GNP/capita
- political accountability - freedom of the press

The data set generated for this class contains several measures for each of these three variables, which will allow you to create a typology of nations by utilizing other indicators of the genotype. Alternatively, you may wish to use the measures of size-development-political accountability in the actual proposition which you are testing. For example, the proposition - As the size of a nation increases, the amount of hostile activity in which it engages will also increase - uses the size variable in a way other than for typing nations. The Rosenau framework also suggests that behavior is influenced by individual, governmental, societal, external, and systematic attributes. Consequently, data representing some of these clusters (individual and systemic influencers are eliminated because of a dearth of data for the time span of this handbook) are included in your data set of 72 variables. Thirty-five of these may be considered input or independent variables; that is, they represent attributes of nations - their size, level of development, degree of political accountability, governmental and societal factors, and external variables (geographic position vis-a-vis the major powers, national attributes in comparison to one's neighbors, and access to the seas). This list of variables follows:

Large				Small			
Developed		Underdeveloped		Developed		Underdeveloped	
Open	Closed	Open	Closed	Open	Closed	Open	Closed
France - 229	Poland - 290	Brazil - 140	Red China - 710	Argentina - 160	Albania - 339	Bolivia - 145	Afghanistan - 700
W. Germany - 255	U.S.S.R. - 345	India - 750	Indonesia - 850	Australia - 900	Bulgaria - 355	China (T.) - 713	Algeria - 615
Italy - 325		S. Korea - 732	Pakistan - 770	Austria - 305	Czechoslovakia	Colombia - 100	Burma - 775
Japan - 740		Mexico - 070	Spain - 230	Belgium - 211	- 315	Costa Rica - 094	Cambodia - 811
U.K. - 200		Nigeria - 475	U.A.R. - 651	Canada - 020	E. Germany - 265	Dom. Rep. - 042	Cameroun - 471
U.S. - 002		Philippines - 840		Cyprus - 352	Hungary - 310	Ecuador - 130	Caylon - 780
		Thailand - 800		Denmark - 390	Rumania - 360	El Salvador - 092	Chad - 483
		Turkey - 640		Finland - 375		Greece - 350	Congo (Kl.) - 490
				Ireland - 205		Guatemala - 090	Cuba - 040
				Israel - 666		Honduras - 091	Ethiopia - 530
				Netherlands - 210		Jamaica - 051	Ghana - 452
				New Zealand - 920		Lebanon - 660	Haiti - 041
				Norway - 385		Malaysia - 820	Iran - 630
				Sweden - 380		Morocco - 600	Iraq - 645
				Switzerland - 225		Panama - 095	Jordan - 663
				Venezuela - 101		Peru - 135	N. Korea - 731
						S. Africa - 360	Laos - 812
						Uganda - 500	Nepal - 790
						Uruguay - 165	Portugal - 235
							Senegal - 433
							S. Vietnam - 817
							Syria - 652
							Tunisia - 616
							Upper Volta - 439
							Yugoslavia - 345
				Iceland - 395**		Burundi - 516**	Mauritania - 435
				Kuwait - 690		Central African	Mongolia - 712
				Luxembourg - 711		Republic - 482	Nicaragua - 093
				Trinidad and		Chile - 155	Niger - 436
				Tobago - 052		Congo (Br.) - 484	N. Vietnam - 816
						Dahomey - 434	Paraguay - 150
						Gabon - 481	Rwanda - 517
						Guinea - 438	Saudi Arabia - 670
						Ivory Coast - 437	Sierra Leone - 451
						Liberia - 450	Somalia - 520
						Libya - 620	Sudan - 625
						Madagascar - 580	Togo - 461
						Mali - 432	Yemen - 678

* Cut-off point = mean
 Mean of pop. = 26,737
 Mean of Dev. = 527,9310
 Mean of open-closed = 438,3678

** Missing data for the political accountability variable

N = 6 N = 2 N = 8 N = 5 N = 16 N = 6 N = 19 N = 25

FIGURE 2. 1963 Genotypic Nation Clusters by Means*



Part III: Variable Definitions and Sources

The data set outlined in Part II has been extracted from a larger data base developed under the CAPE (Comparative Analysis of Policy Environments) Project. This latter project has been conducted by the Behavioral Sciences Laboratory at the Ohio State University. The CAPE Project data base consists of a data matrix with row and column dimensions of 116 x N. The number 116 represents those national political units that were members of the United Nations in 1967 or that were excluded from membership because of their divided status (such as North and South Vietnam). The number N represents the number of variables in the data set at a given moment, a list which is constantly expanding as user needs continue to change.

The student data set utilized in this handbook contains 72 variables. Thus the data matrix is 116 x 72.

The data set has been placed on IBM punched cards which each student will receive. A standard punched card contains 80 columns, in each of which can be placed one unit of information. Information or data are arranged in fields; thus, if each datum can be placed in one field, a maximum of 80 units of data may be placed on one card. Since the data set is composed of 72 variables, it is possible to place all of the information for each nation on one card. The complete data set has been punched on 116 cards, each of which represents one nation of our sample.

The one-field code values assigned to the 72 variables are given in the first 72 columns. Three-letter and three-digit codes for each country are given in columns 74-76 and 78-80 respectively. For example, the card representing the data for the United States contains the letters "USA" in columns 74-76 and the numbers "002" in columns 78-80.

The country identifications and definitions of variables 1-10 are given below. (Definitions of variables 11-72 should be obtained from the author.)

002	USA	United States	325	ITA	Italy	600	MOR	Morocco
020	CAN	Canada	339	ALB	Albania	615	ALG	Algeria
040	CUB	Cuba	345	YUG	Yugoslavia	616	TUN	Tunisia
041	HAI	Haiti	350	GRC	Greece	620	LYB	Libya
042	DOM	Dominican Republic	352	CYP	Cyprus	625	SUD	Sudan
051	JAM	Jamaica	355	BUL	Bulgaria	630	IRN	Iran
052	TRI	Trinidad and Tobago	360	ROM	Romania	640	TUR	Turkey
070	MEX	Mexico	365	USR	USSR/Russia	645	IRQ	Iraq
090	GUA	Guatemala	375	FIN	Finland	651	UAR	United Arab Republic
091	HON	Honduras	380	SWD	Sweden	652	SYR	Syria
092	ELS	El Salvador	385	NOR	Norway	660	LEB	Lebanon
093	NIC	Nicaragua	390	DEN	Denmark	663	JOR	Jordan
094	COS	Costa Rica	395	ICE	Iceland	666	ISR	Israel
095	PAN	Panama	432	MLI	Mali	670	SAU	Saudi Arabia
100	COL	Colombia	433	SEN	Senegal	678	YEM	Yemen
101	VEN	Venezuela	434	DAH	Dahomey	690	KUW	Kuwait
130	ECU	Ecuador	435	MAU	Mauritania	700	AFG	Afghanistan
135	PER	Peru	436	NIR	Niger	710	CHN	Red China
140	BRA	Brazil	437	IVO	Ivory Coast	712	MNW	Mongolia
145	BOL	Bolivia	438	GUI	Guinea	713	CHI	China (Taiwan)
150	PAR	Paraguay	439	UPP	Upper Volta	731	KOR	North Korea
155	CHL	Chile	450	LIB	Liberia	732	KOS	South Korea
160	ARG	Argentina	451	SLI	Sierra Leone	740	JAP	Japan
165	URU	Uruguay	452	GHA	Ghana	750	IND	India
200	UNK	United Kingdom	461	TGO	Togo	770	PAK	Pakistan
205	IRE	Ireland	471	CAF	Cameroun	775	BUR	Bur
210	NTH	Netherlands	475	NIG	Nigeria	780	CEY	Ceylon
211	BEL	Belgium	481	GAB	Gabon	790	NEP	Nepal
212	LUX	Luxembourg	482	KEN	Ken. African Rep.	800	TAI	Thailand
220	FRN	France	483	CHA	Chad	811	CMB	Cambodia
225	SWZ	Switzerland	484	CON	Congo (Br.)	812	LAO	Laos
230	SPN	Spain	490	COP	Congo (Ki.)	816	VTR	North Vietnam
235	POR	Portugal	500	UGA	Uganda	817	VTS	South Vietnam
255	GMW	West Germany	516	BUR	Burundi	820	MAL	Malaysia
265	GMF	East Germany	517	RWA	Rwanda	840	PHI	Philippines
290	POL	Poland	520	SOM	Somalia	850	IND	Indonesia
305	AUS	Austria	530	ETH	Ethiopia	900	AUL	Australia
310	HUN	Hungary	560	SAY	South Africa	920	NEW	New Zealand
315	CZE	Czechoslovakia	580	MAG	Malagasy			

TABLE 2. Country Identification Key

Variable Definitions

Variable 1: Total Population

Variable Definition: "The total population of a country is conventionally described as de facto or de jure. A true de facto or present-in-area concept implies that all persons physically present in the country - residents and non-residents alike - have been counted in the local area where they were found at the time of the census. The de jure count, in contrast, comprises all persons who usually reside in the area, irrespective of where they might happen to be at the time of the census. Simple as these concepts appear, strict conformity to either of them is rarely found In an effort to provide better information for constructing world and regional population aggregates from the results of censuses taken around 1950, the Population Commission of the United Nations recommended a "modified de facto" tabulation of the total population in addition to any other total used for national purposes. This same concept was included in the Principles and Recommendations for National Population Censuses designed to provide guidance for the taking of National Population Censuses designed to provide guidance for the taking of the 1960 cycle of censuses. In the 1960 recommendations, the new total was called "international conventional total", and it was defined as "the total number of persons present in the country at the time of the census, excluding foreign military, naval and diplomatic personnel and their families located in the country but including military, naval and diplomatic personnel of the country and their families located abroad, and merchant seamen resident in the country but at sea at the time of the census.... Since the computed total was called for in addition to any other total used for national purposes, there is no expectation that it would necessarily have been used in the detailed census tabulations. Therefore, beginning with the 1963 Demographic Yearbook, the assumption that all census results refer to the modified de facto population was abandoned, and the modification has been described only where it is known to have been made." "Unless otherwise noted, population figures are present-in-area estimates for the present territory."

The data is in thousands of persons. The year of the data is 1963.

Citation: Demographic Yearbook 1966, 185th Issue, Statistical Office of the United Nations, Department of Economic and Social Affairs, New York, 1967.

Variable 2: Total Land Area

Variable Definition: "Unless otherwise specified, all of these figures are assumed to represent total area, that is, they comprise the land area and inland waters, excluding only polar regions and some uninhabited islands. Inland waters are assumed to consist of major rivers and lakes In this yearbook, area is given in square kilometers, the conversion from square miles (if required) having been accomplished by equating 1 square mile to 2.589998 square Kilometers."

Citation: Demographic Yearbook 1963, Statistical Office of the United Nations, Department of Economic and Social Affairs, New York, 1964.

Variable 3: Total GNP

Variable Definition: GNP in millions of U. S. dollars. No definition given in source. For a definition of GNP refer to Sprecher GNP (Cape Variable 053).

The data is for the year 1963.

Citation: "Estimates of GNP, 1963," Agency for International Development, Report Control #137, (01), Statistics and Reports Division, February 19, 1965.

Variable 4: KWH of Electrical Production

Variable Definition: KWH of electrical production for 1963. "...All the figures on capacity and production represent combined totals for electrical utilities and industrial establishments having generating facilities for providing all or part of their own requirements."

The data is in millions of KWH.

Citation: World Power Data, 1964, Bureau of Power, Federal Power Commission, Washington, May 1965.

Variable 5: GNP Per Capita

Variable Definition: GNP per Capita. No definition given in source. The data is in U. S. dollars per capita. The year of data is 1963.

Citation: Statistics and Reports Division, Agency for International Development, February 19, 1965.

Variable 6: Energy Consumption Per Capita

Variable Definition: A nation's total energy consumption divided by its total population. All data are for 1963. The unit of measure is metric tons of coal equivalents per capita.

Citation: Total Energy Consumption: World Energy Supplies 1962-1965, Department of Economic and Social Affairs, Statistical Office of the United Nations, New York, 1967.

Variable 7: Agricultural Workers As % of Total Economically Active Population

Variable Definition: "Agricultural population for the purpose of this table may be defined as all persons actively engaged in agriculture, forestry, hunting and fishing for a livelihood, that is to say, persons actively engaged in agriculture, forestry, hunting and fishing and their non-working dependents.... In general, the economically active population is defined as all persons engaged in an economic activity, whether employers, own-account workers, salaried employees, or unpaid workers assisting in the operation of a family farm or business. Similarly the population economically active in agriculture includes all economically active persons engaged principally in agriculture, forestry, hunting and fishing.... This general definition differs somewhat from country to country. In some countries, for example, the estimates are based on data relating to all persons reporting an occupation, whether or not they were actually working at the time of the census or survey; in others, on data regarding persons actually employed during a specific short period, unemployed persons seeking work being excluded. Some countries report information on economic activity for persons of all ages, others only for persons of specified ages, e.g., 14 years of age and over."

The data are expressed as a percentage and are for the year 1965.

Citation: Production Yearbook 1966, Vol. 20, Food and Agriculture Organizations of the United Nations, Rome, 1967.

Variable 8: KWH Per Capita

Variable Definition: KWH per capita for 1963. The data were calculated within the source. "...All the figures on capacity and production represent combined totals for electrical utilities and industrial establishments having generating facilities for providing all or part of their own requirements."

Citation: World Power Data, 1964, Bureau of Power, Federal Power Commission, Washington, May 1966.

Variable 9: Current Electoral System

Variable Definition: A = Competitive (no party ban, or ban on extremist or extra-constitutional parties only); B = partially competitive (one party with 85% or more of legislative seats); C = non-competitive (single-list voting or no elected opposition).

Citation: A Cross-Polity Survey, Arthur S. Banks and Robert B. Textor, The M.I.T. Press, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Variable 10: Freedom of the Press

Variable Definition: Freedom of the press scores - 100 (PICA index + 4). The PICA index is based on 23 indices of freedom of the press. The strength of the indices were judged by both native and non-native judges in response to a questionnaire. The

scores were averaged separately for the native and non-native judges. Where there were disagreements of more than 6 per cent between the averages, only the non-native averages were used. Where there was less than a 6 per cent disagreement, the average of the two was used for the PICA index.

Citation: PICA: Measuring World Press Freedom, Lowenstein, Freedom of Information Center, University of Missouri, 1966.

Part IV: Variable Coding Scheme

In this section the coding scheme and specific categories for each variable will be revealed. All of the variables in the data set distributed to you are in ordinal measures. That is, each variable has been divided into categories and assigned a number in such a fashion that each number stands in a definite relationship to every other number; its quantity is greater than or less than that represented by the remaining numbers. For example, variable 15 represents the current status of the legislature's effectiveness. This variable is divided into 4 categories, numbered in the following fashion:

- 1 = fully effective
- 2 = partially effective
- 3 = largely ineffective
- 4 = wholly ineffective

As you can observe from the category names, it is impossible to ascertain the exact distance between each category (as you could with a variable which revealed the number of times the legislature overrules the executive's veto, for example). You only know that if a nation has been assigned the code value "2", the effectiveness of its legislature is greater than that of a nation which has been assigned the value "3", but less than that nation which is given the value "1."

In their original format, some of the 72 variables selected for the data set were ordinal in nature (8) while most were comprised of interval-level data (64). All interval-level measures have been reduced to ordinal data for the purposes of this manual. This was done for two reasons. First, since a student will not become familiar with statistical techniques requiring interval data (such as regression analysis) until later in the course, ordinal measures will suffice for the earlier portion of the course.

The second reason is convenience. In order to keep the number of cards in the student's data deck to a small number (one card per country) it is necessary to fit 72 variables on one card allocating one column per variable. Since interval measures require more than one column to express the value of the datum, ordinal measures are chosen instead.

Data Reduction Method. The following steps were employed to reduce the data to the ordinal level of measurement:

- a. The interval-level variables to be included in the analysis deck were "pulled out" of the larger CAPE data files and standardized with a mean = 100 and a standard deviation = 10. The distributions for each variable had already been normalized (mean=median=mode) with appropriate data transformations. (These terms are explained in the Burgess-Peterson laboratory manual and will be discussed fully in class).
- b. The range of the standardized data was computed for each variable.
- c. Each range was divided into ten equal intervals (deciles).
- d. Ordinal values from 0 to 9 were assigned to each decile with 0 denoting the lowest decile and 9 denoting the highest.
- e. The values for each country on every variable were placed in the appropriate decile.
- f. The decile number was assigned as the ordinal code for each datum.

The ordinal values for variables 1-10 are given below. Values for variables 11-72 can be obtained from the author.

Ordinal Values for Variables

VARIABLE 1: TOTAL POPULATION

0 = 74.87 to 80.47
1 = 80.48 to 86.08
2 = 86.09 to 91.69
3 = 91.70 to 97.30
4 = 97.31 to 102.91
5 = 102.92 to 108.52
6 = 108.53 to 114.13
7 = 114.14 to 119.74
8 = 119.75 to 125.35
9 = 125.36 to 130.96

VARIABLE 4: KWH OF ELECTRICAL PRODUCTION

0 = 81.10 to 85.48
1 = 85.49 to 89.87
2 = 89.88 to 94.26
3 = 94.27 to 98.65
4 = 98.66 to 103.04
5 = 103.05 to 107.43
6 = 107.44 to 111.82
7 = 111.83 to 116.21
8 = 116.22 to 120.60
9 = 120.61 to 124.99

VARIABLE 2: TOTAL LAND AREA

0 = 72.27 to 77.70
1 = 77.71 to 83.14
2 = 83.15 to 88.58
3 = 88.59 to 94.02
4 = 94.03 to 99.46
5 = 99.47 to 104.90
6 = 104.91 to 110.34
7 = 110.35 to 115.78
8 = 115.79 to 121.22
9 = 121.23 to 126.66

VARIABLE 5: GNP PER CAPITA

0 = 81.38 to 85.28
1 = 85.29 to 89.19
2 = 89.20 to 93.10
3 = 93.11 to 97.01
4 = 97.02 to 100.92
5 = 100.93 to 104.83
6 = 104.84 to 108.74
7 = 108.75 to 112.65
8 = 112.65 to 116.56
9 = 116.57 to 120.47

VARIABLE 3: TOTAL GNP

0 = 83.10 to 87.84
1 = 87.85 to 92.59
2 = 92.60 to 97.34
3 = 97.35 to 102.09
4 = 102.10 to 106.84
5 = 106.85 to 111.59
6 = 111.60 to 116.34
7 = 116.35 to 121.09
8 = 121.10 to 125.84
9 = 125.85 to 130.58

VARIABLE 6: ENERGY CONSUMPTION PER CAPITA

0 = 77.90 to 82.10
1 = 82.11 to 86.31
2 = 86.32 to 90.52
3 = 90.53 to 94.73
4 = 94.74 to 98.94
5 = 98.95 to 103.15
6 = 103.16 to 107.36
7 = 107.37 to 111.57
8 = 111.58 to 115.78
9 = 115.79 to 119.99

VARIABLE 7: AGRICULTURE WORKERS AS % OF TOTAL ECONOMICALLY ACTIVE POPULATION

0 = 81.20 to 85.36
1 = 85.37 to 89.53
2 = 89.54 to 93.70
3 = 93.71 to 97.87
4 = 97.88 to 102.04
5 = 102.05 to 106.21
6 = 106.22 to 110.38
7 = 110.39 to 114.55
8 = 114.56 to 118.72
9 = 118.73 to 122.89

VARIABLE 8: KWH PER CAPITA

0 = 74.78 to 79.30
1 = 79.31 to 83.83
2 = 83.84 to 88.36
3 = 88.37 to 92.89
4 = 92.90 to 97.42
5 = 97.43 to 101.95
6 = 101.96 to 106.48
7 = 106.49 to 111.01
8 = 111.02 to 115.54
9 = 115.55 to 120.07

VARIABLE 9: CURRENT ELECTORAL SYSTEM

- 1 = Competitive (no party ban, or ban on extremist or extra-constitutional parties only).
- 2 = Partially competitive (one party with 85% or more of legislative seats).
- 3 = Non-competitive (single-list voting or no elected opposition).

VARIABLE 10: FREEDOM OF THE PRESS

0 = 82.90 to 86.58
1 = 86.59 to 90.27
2 = 90.28 to 93.96
3 = 93.97 to 97.65
4 = 97.66 to 101.34
5 = 101.35 to 105.03
6 = 105.04 to 108.72
7 = 108.73 to 112.41
8 = 112.42 to 116.10
9 = 116.11 to 119.79

THE CREATION AND DIFFUSION OF INNOVATIVE USES OF THE
COMPUTER IN SOCIOLOGY EDUCATION

Ronald Stiff
Daniel Vanderportaele
Illinois Institute of Technology
Chicago, Illinois 60616
Telephone: (312) 255-9600

This seems an appropriate time to evaluate the creation and diffusion of innovative uses of the computer in sociology education. It is the time to ask "to what extent have we been innovative" and "to what extent have these innovations diffused through the educational community"? Toward these goals we propose a classification scheme suggesting several possible innovative approaches. Published innovations are compared to this scheme to evaluate the extent to which we have exhausted these innovative possibilities. To evaluate the feasible diffusion of these innovations a brief analysis of the potential sites for adoption of computer based educational materials is developed. The likely actual diffusion of these computer enriched educational materials is matched against these potential adoption sites. In concluding several suggestions for encouraging creation and diffusion of innovative computer based educational materials are made.

Some portions of this paper are based on experiences gained during the initial three years of "A Cooperative Venture in Curriculum Development Based on a Regional Computer Network at Illinois Institute of Technology(1)". Ronald Stiff was Project Manager for the project which involved computer based curriculum development in seven academic disciplines at ten midwestern colleges supported by the I.I.T. UNIVAC 1108. Daniel Vanderportaele served as Sociology Curriculum Development leader.

Classification of Innovations

Numerous classifications of computer uses in education have been proposed. Luehrmann's (1971) ten nodes is one of the more useful:

1. Management of instruction and materials.
2. Administration of drill and practice sessions.
3. Conversations and dialogs.
4. Large data-base inquiry systems.
5. Simulation.
6. Problem solving.
7. Laboratory data analysis.
8. Laboratory data acquisition.
9. Control of experiments.
10. Production of graphics, movies and other media."

Although Luehrmann is a physicist, these nodes are all suitable for use in education in sociology. (Laboratory, of course, has a somewhat different meaning to sociologists than to the physicists.) The scheme, however, can be made more useful by consideration of the substance of sociology.

Although methodology and theory are recognized as interdependent, for instructional purposes it is often useful to treat each independently. Additionally, at times we deal with concepts, at times with empirical data. Based on these factors we have developed the following substantive classification for instructional uses of computers in sociology:

	<u>Methodology</u>	<u>Theory</u>
<u>Conceptual</u>	(A) Programs which teach methodological concepts and the logical basis of scientific inquiry.	(B) Programs which aid in theory construction through application of symbolic logic and the creation of models.
<u>Data Specific</u>	(C) Programs which analyze data and produce outputs requiring interpretation.	(D) Programs to develop "causal models" or test theory through sensitivity analysis.

472

Our sociology specific scheme may be superimposed on Luehrmann's to produce up to forty possible computer based educational strategies. Of course, several of these make little sense (e.g. 7A, 8B, etc.) but most provide useful suggestions for educational innovations. For example, large scale data bases may be developed containing both data and propositional inventories. The propositional inventories may be used to demonstrate the concepts of theory building (4B). Data may then be used to develop causal models testing these theories (4D). Anyone who has attempted to communicate the concepts of internal and external validity in experimental design (e.g. regression artifacts, history, etc.) would be likely to welcome a conversational program demonstrating these methodological concepts (3A). Computer based dialogs may be developed to instruct the student in strategies of interviewing (3A). Other possibilities are suggested by this scheme but it is not our goal to be exhaustive, but to demonstrate the suggestive potential provided. This scheme also provides a standard for evaluating the extent to which innovative possibilities have been explored. Next, however, we consider the potential diffusion of computer based educational materials in sociology.

Possible Diffusion of These Innovations

An educational innovation in sociology could originate at one or more of the over 800 colleges and universities offering an undergraduate major in sociology and diffuse to some portion of the remainder. Since methodology courses are the most obvious settings to introduce the computer we might expect initial innovation at schools requiring research methods or statistics for their majors(2). In addition to having a potential need for introducing computer innovation, the sociology faculty must have access to a computer facility. Hamblen (1971) reports that about half of the 2,500 institutions of higher education (including junior colleges) have some computer facility. We would expect a reasonable positive correlation between schools requiring methodology courses of their majors and schools having a computer facility. Therefore, it does not seem unreasonable to estimate that over 200 colleges have both course requirements and computer facilities permitting perhaps even encouraging, innovations in computer supported sociology education.

An argument can be made that computer facilities at many colleges are not substantial enough to perform effectively the most obvious use, statistical, analysis of large data bases. (Strategy 3C). The small computers (e.g. IBM 1130) available at many colleges seldom have sufficient processing speed or suitable statistical libraries for meaningful data analysis. Although this may be true in part, this problem is virtually eliminated at small colleges who make remote use of larger computers at universities, in their region. Since 1968 the National Science Foundation, Office of Computing Activities, has provided support of regional computing activities whereby universities with major computer hardware and software act as a resource for several participating schools. Through 1971, 23 regional networks had been established, assuring computing resources for 228 colleges and universities. Assuming half these schools offered sociology as a major at least 100 colleges and universities have had access to a university computer center and presumably some rewards would be gained by making use of this resource.

In summary, we have demonstrated that there are numerous ways in which the computer may be used to enrich sociology education. Also, there appears to be at least 100, if not several hundred, colleges and universities which have both computer resources and course requirements encouraging innovative use of the computer in undergraduate education. Relative to this theoretical mosaic what has been accomplished?

What Has Been Accomplished

Published reports of computing activities supporting undergraduate sociology education suggest that students at large schools with large computers are analyzing data bases in methodology courses. Returning to the original classification scheme this is strategy 4C. Pedagogical differences in data base analysis include, having students gather their own data (O'Kane: 1970), reaching hundreds of students (Anderson and McTavish, 1970), providing rich data bases and inexpensive conversational analysis (Meyers, 1970 and Davis, 1971), instant turnaround (Kreider, et. al., Sim, et. al., 1971), and use in an underdeveloped area with a PDP/10 computer (Wildgen; 1971). In searching for uses other than data specific methodology using large data bases (4C) our rewards are few.

Vargus and White (1971) have used simulation programs in an Urban Affairs course to demonstrate informal neighborhood social integration (Strategy 5D). Vandeportaele has developed models of three-man dominance and human interaction (Homan's propositions) that are problem-solving demonstrations of the concepts of theories (Strategy 6B) and a computer simulation of the growth of urban areas (Strategy 5D). Nicholas Mullins, Indiana University, has developed conversational programs demonstrating concepts in theory construction (Strategy 3B) and causal modeling (Strategy 3D).

The majority of the reported innovations applying computers to undergraduate sociology education have considered varieties of data base analysis in methodology courses. Of the up to

473

39 other possible strategies a great deal of creative potential remains. The sources of these innovations reveal additional opportunities for creation and diffusion of innovations.

It is instructive to consider the sources of the papers accepted at the first two conferences on computers in undergraduate education. Sociology papers were written by authors at ten schools with a median student population of 10,000. None of these schools were remote participants in the N.S.F. funded regional computer networks, although three provided the network central computing resources. Although there are numerous schools with under 2,000 students only one paper originated from a school of this size. With the exception of Francis, McGinnis, and Schnell's (1970) discussion of the future desirability of reaching small colleges none of the papers considered the diffusion of computer innovations to other schools.

There is a strong suggestion, although neither proof nor explanation, that modest size colleges are making virtually no use of the computer in undergraduate sociology education. Our experiences at Illinois Institute of Technology as an N.S.F. funded regional computer activity provide partial explanations for poor diffusion of the computer innovations to small colleges.

During the initial three years of the project an average of one sociologist per campus participated in the sociology curriculum development project(3). Each participating faculty member averaged about 20 minutes of computer time annually, while assisting 20 students. During this period we provided about 50 computer programs for data base analysis (Strategy 7C) and several other strategies. These programs were adequately designed and documented, but certainly not as highly polished as most text materials and teacher's manuals(4). Virtually none were adopted by our participating faculty with the exception of some minor statistical routines(5). At no time did we achieve any significant use of these programs. Therefore, this year we have turned totally to the use of statistics programs to analyze data bases (Strategy 4C). We are not aware of any other substantial adoptions of computers by sociology faculty at small schools. There are certainly more than we list here.

We attribute this lack of success to several conditions, external and internal to these schools. External to the college is the general failure of sociologists to create computer applications which are suitable for use in undergraduate education. "Computer assisted" sociology, beyond data analysis, does not seem to be one of the stronger forces in sociology today. Secondly, very few PhD graduates in sociology are trained in computer usage other than data analysis. Thirdly, with the exception of Dartmouth's project IMPRESS (a system which is quite difficult to export due to its programming specifically for Dartmouth's computers), there has been very little research funding for development of computer enriched sociology teaching materials and systems.

Within the college there are additional problems. College administrations are seldom aggressively encouraging faculty development. Faculty at many schools have heavy teaching loads and low salaries, often leading to a need for supplementation through summer teaching. Little time is available for development and utilization of new skills. This contributes to rather fixed and frequently out-dated curriculum in which adoption of the computer has a very low priority. This may result from other material having an absolutely higher priority or the computer providing absolutely too much pain. Additionally, there is always the fear of alienating students by exhibiting personal limitations or teaching "non-humanistic" sociology.

In summary, of all possible innovative strategies for using computers in sociology education few have been exploited. Of all possible sources for creating innovations and sites for diffusion few have responded. Evaluated in terms of opportunity we can be optimistic about the possibilities that remain open. But how can we take advantage of the current "crisis" in computer enriched sociology education?

In Conclusion - How Can We Encourage Innovation and Diffusion

We have made an argument for two needs or opportunities in the creation of computer innovations and the diffusion of these innovations. The initial creation of several innovations should not be difficult. The scheme for categorizing educational uses of the computer presented in this paper suggests several new directions. Some teaching materials may be developed without external research funding, although funds directed into the undeveloped uses and multiple researcher and school cooperative ventures are likely to produce a substantial marginal return. Funding agencies should provide incentives for directing efforts toward developing innovative materials in the numerous underdeveloped strategies(6).

A greater barrier to widespread computer usage in sociology education is the problem of diffusing existing innovations. The only programs achieving reasonable wide spread distribution are data analysis systems(7). Statistical package for the Social Sciences (S.P.S.S.) is the best documented and most widely diffused of these programs. We have found teaching sub-sets of S.P.S.S. quite satisfactory for use with seniors and graduate students. When allowing optional use in a methodology class, ten of 34 students elected to use S.P.S.S. at a cost of \$550. This cost is excessive for a single course and can only be marginally justified if it is the

student's only exposure to computer analysis or it is costed over the total class size at \$16 per student. These systems are often well documented and provide sufficient flexibility for an instructor to adapt them to his own knowledge and style of teaching. In this manner analysis systems close the gap between innovation and diffusion by reducing the need of the sociology faculty to become computer knowledgeable, to develop texts, or to reprogram.

This suggests opportunities for speeding diffusion of computer innovations; the development of well documented, pedagogically flexible programming systems, requiring virtually no computer knowledge by the consumer, inexpensive to use, and transportable over a wide variety of computer systems. Thus a theory building computer enriched teaching unit should permit a variety of methodological and theoretical perspectives. Original creation of transportable programming systems of this sort will be quite costly. The goal of such projects, however, is wide spread diffusion and the cost per adopting school and student reached could be quite low. We do not believe these systems can be developed without funding agencies encouraging team efforts and high standards for the initial development.

At the same time these program systems are being developed sociology faculty should be trained in the use of the computer in education. In our regional network we have demonstrated that this training is not effective on a part-time basis. Full time, four to eight week summer institutes modeled on those sponsored by the National Science Foundation seem the most effective strategy (8). Summer institutes can produce computer capable sociologists at a cost of about \$3,000 per sociologist. If we assume that a faculty member is able to effectively communicate his skills to 200 students in a five year period following completion of the summer institute an expenditure of \$15 per student results. Two institutes could produce 50 to 80 computer competent faculty members annually. The summer institute setting eliminates some of the institutional constraints on faculty development and provides an excellent opportunity for faculty at a variety of schools to exchange knowledge over several weeks. It may be desirable to select participants on a regional basis to encourage communication following the institute. If two or more institutes are held in a given year, instructional resources should be shared to provide a variety of viewpoints.

In conclusion, we believe that those of us who support this annual conference should restructure conference sessions to encourage innovation and diffusion of the computer in sociology education. One session should be devoted to innovative computer enriched teaching materials. Development of uses other than analysis of data bases should be encouraged. A second session should be devoted to the diffusion of computer enriched teaching materials. Experiences at schools of various sizes and character should be encouraged. Instances of materials being adopted at one or more schools should be emphasized. By demonstrating what has been done, these conferences demonstrate what could be done.

FOOTNOTES

1. N.S.F. Grant CJ-281: Peter G. Lykos, Principal Investigator 1968-71.
2. Approximately 400 and 200 respectively if Bates and Reid (1971) are generalized.
3. Almost none of the participants received release time from their normal nine to twelve hour teaching load.
4. These programs were developed as Computer Enriched Teaching Units (CETU) detailing the teaching problems, computer programs required, consumer documentation, and text material to illuminate the principles treated in the programs. CETU programs were entered into our Cooperative Program Exchange Service (COPES) library to facilitate distribution to other schools.
5. Several programs were exported to the University of Texas and the North Carolina regional networks where they reportedly received some use.
6. The computational laboratory concept proposed by Lykos (1970) should be considered in some detail by sociologists since it may provide cost-effective support for most of the innovative strategies proposed here.
7. See Anderson; 1971 for a survey of data analysis software.
8. Unfortunately these programs have been cut back greatly in 1972 and none of the existing institutes are helpful to sociologists in developing computer abilities. Weinstock, of I.I.T. provides an excellent model of a summer institute for physicists.

REFERENCES

1. Anderson, Ronald E., "A Survey of Application Software For Social Data Analysis," Proceedings of the Second Annual Conference on Computers in Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.
2. Anderson, Ronald E., "A Survey of Application Software For Social Data Analysis Instruction," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, 1971.
3. Anderson, Ronald E., and McFavish, Donald C., "Sociology, Computers and Mass Undergraduate Education," Proceedings of a Conference on Computers in the Undergraduate Curricula, Iowa City, Iowa, The University of Iowa, 1970.
4. Bates, Alan P. and Sam Titus Reid, "The Sociology Major in Accredited Colleges and Universities," The American Sociologist, Vol. 6, pp. 243-249, 1971.
5. Davis, James A., "Using IMPRESS to Teach Sociology," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.
6. Denk, Joseph R., "Curriculum Development of Computer Usage in North Carolina," Proceedings of a Conference on Computers in the Undergraduate Curriculum, Iowa City, Iowa, The University of Iowa, 1970.
7. Francis, J., McGinnis, R., and Schnell, M., "Computer-Based Instruction in the Social Sciences - an Experimental Course in Research Methods," Proceedings of a Conference on Computers in the Undergraduate Curriculum, Iowa City, Iowa, The University of Iowa, 1970.
8. Hasbun, John W., "Using Computers in Higher Education: Past Recommendations, Status, and Needs," Communications of the ACM, Vol. 14, No. 11, pp. 709-712, 1971.
9. Kreider, Glen D., Sia, Francis H., and Williams, Anthony V., "Instant Turnaround in Instructional Computing - Some Examples From the Social Sciences at the Pennsylvania State University," Proceedings of a Conference on Computers in the Undergraduate Curriculum, Iowa City, Iowa, The University of Iowa, 1970.
10. Luehrmann, Arthur W., "Dartmouth Project COEXIST. The Computer Qua Computer," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.
11. Lykos, Peter G., "After Token Academic Computer Use - Then What?" Educational Media November, 1970.
12. Meyers, Edward D., "IMPRESS and Undergraduate Education in the Social Sciences," Proceedings of a Conference on Computers in the Undergraduate Curriculum, Iowa City, Iowa, The University of Iowa, 1970.
13. O'Kane, James H., "The Application of Empirical and Computer Techniques to Undergraduate Sociology Research Courses," Proceedings of a Conference on Computers in the Undergraduate Curriculum, Iowa City, Iowa, The University of Iowa, 1970.
14. Sia, Francis H., Gordon F. DeJong, Glen D. Kreider, and David Kaufman, "Data Analysis for Sociology Undergraduates: Innovations of Instant Computation," The American Sociologist, Vol. 6, No. 2, pp. 153-157, 1971.
15. Sia, Francis H., Laurence S. Rosen, and Glen D. Kreider, "Instant Turnaround and Conversational Computing as Instructional Tools in the Social Sciences," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.
16. Vargus, Brian S., and Douglas White, "Report on an Attempt to Utilize Computers in Urban Affairs Education," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.
17. Wildgen, John K., "Computers and Undergraduate Training in an Underdeveloped Area: The Case of Louisiana State University in New Orleans," Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Hanover, New Hampshire, The University Press of New England, 1971.

POISSON - A Daughter of Dartmouth's IMPRESS
Has Been Born in the Environment of
IBM Time-Sharing

Joseph R. Deak
North Carolina Educational Computing Service
Research Triangle Park, North Carolina 27709
Telephone: (919) 549-8291

Introduction

The famed survey analysis system of Dartmouth College, Project IMPRESS [1], has been moved to only two computer centers outside of Dartmouth's Kierit Center: to the Naval Academy (with a near identical system) and the one-day delivery of a disk, and to an IBM 360 Model 75 at the Triangle Universities Computation Center, TUCC, in North Carolina. POISSON, Package of Instructional Social Surveys of North Carolina, was born at TUCC in a delivery closer to nine months that involved no ease to the birth pains such as the shipping of disks or tapes. The birth of POISSON provides not only a new system for survey analysis to people time-sharing on IBM 360 or 370 systems but also a delineation of parameters involved in the "transportability" process of interest to all who are faced with this problematic area [2]. Transporting tutorial developments and pedagogical advances along with computer programs to an entirely different time-sharing environment is both a financial necessity and a sine qua non condition for curriculum usage of the computer to the vast majority of social scientists who are yet unable to produce these systems themselves.

IMPRESS has a remarkable history and has been well documented in this series of conferences [3,4,5,6]. Not only has the system developed to be a powerful tool for undergraduate instruction in survey analysis but the student involvement in its evolution has provided, at least theoretically, the base on which all educational development should ride. A student refined system operating on real data and pointing toward significant research on the undergraduate level offered an optimistic mix of educational design criteria available in social survey systems. The North Carolina Educational Computing Service (NCECS), an NSF funded network involved in curriculum development, produced POISSON in one year with the goal of retaining as much of the design criteria and current power of IMPRESS as possible. POISSON has run on an IBM 360 Model 75 and is now operational on a 370/165 at TUCC. The entirely new system, POISSON, involved transporting NO PROGRAMS but a vast quantity of concepts, the documentation of which "transportability" is unavailable, to the author's knowledge, for any educational system. Limitations of the daughter package are also essential to communicate to those IBM users who desire a true picture of the availability of POISSON.

Transportability

What may seem remarkable to the potential user of POISSON is that a survey analysis system composed of programs and data bases was transported without any programs or data bases from Dartmouth being involved. Not even an algorithm was adopted from the IMPRESS system. A specific rationale for starting from scratch (what would seem to be a waste of time) will be treated in the next section dealing with the difference in the computer systems at the parent and daughter sites. Transportability involves moving social science, survey analysis, and pedagogical methodology and not programs and data bases, strange as this may seem to the desperate searcher of software. There are many super systems to do survey analysis [5] but few with all the necessary parameters for undergraduate education. The fact that few of these super systems are used in undergraduate education coupled with the historical truth that there is no money in the sales of software should indicate why transportability doesn't involve programs. There is no money in software because it is downright easy to program concepts once the difficult work is done - building the concepts. POISSON is a witness to this fact.

Then why move IMPRESS? The survey analysis system of Dartmouth is the culmination of a billion dollars [7] worth of sociology, political science, survey analysis, pedagogy, sophistication through development, system design, and student interaction to mention a few of the merits. NCECS has spent less than \$10,000 in building POISSON. "Transportability" of the system can be categorized as follows:

Survey Analysis
Social Science
Data Set Construction
Code Books
System Development
Tutorial Pedagogy
Student Feedback

477

Survey Analysis: Among other founders of survey analysis in undergraduate education, James A. Davis (formerly chairman of the Sociology Department of Dartmouth) has provided a theoretical and pedagogical foundation upon which IMPRESS was built[8]. The approach has a philosophical basis which projects that a student who is taught one non-parametric statistic (the gamma) very well can do significant analysis and hypothesis testing on real data if he has a system by which he can reach real data without getting sidetracked by computers and their jargon. To import IMPRESS is to import this philosophy. Davis, Meyers, and a student from Dartmouth gave two seminars to social scientists of North Carolina before the environment was considered prepared for this philosophy. Davis' book[8] has been adopted by at least one participating university, indicating the import of the philosophy itself.

Social Science: Any data set worth its salt is the result of good methodology and contains too many variables for undergraduate consumption, interest, or even relevancy (to say nothing of systems limitations). The largest IMPRESS data set, PRES68, involves only 109 out of 550 possible variables. Subsetting these variables for undergraduate relevancy is social science and a significant factor for consideration in transportability. The first two data sets of POISSON borrowed this work in toto (as well as several other IMPRESS concepts).

Social science is also involved in building standard dichotomies and standard groupings for each variable in order to provide easy first level analysis. The product of this work by social scientists at Dartmouth is transportable.

A published primer[9] for the use of IMPRESS contains social science as well as systems information. NCECS has also published a primer[10] which was not possible without the Dartmouth version.

Data Set Construction: Since IMPRESS data sets require an "inversion" of the original survey from respondent-oriented records to variable-oriented records, the record format resulting from the inversion is an importable item. System differences require reconstruction but the contents of the inverted records are that much easier to produce once available. Further, the problems in selecting educationally relevant data sets are reduced if those used in IMPRESS are adopted. These problems obviously involve the availability of the data sets in the public domain or in some consortium arrangement (such as ICPR[11]) - something to consider!

Codebook: Significant survey analysis requires a reliable codebook for each data set, the production of which is most costly. Since 30 such codebooks are available from Dartmouth at a low cost (about \$2.00 each) the use of these seems warranted at least for the data sets of POISSON at use in IMPRESS. In starting up, POISSON adopted two IMPRESS data sets and their corresponding codebooks: PRES68 (Presidential Election Survey of 1968 - SEC) and ETHNO (Murdoch's Ethnographic Atlas Ethnography Version). The transportability of codebooks was therefore essential to the whole process.

System Development: The evolved IMPRESS system was transportable by careful fulfillment of the systems requirements on IBM time-sharing. With only the output of IMPRESS runs required, the years of refinement were translated into the POISSON version. Although POISSON has not as yet the full power of IMPRESS, the open-ended nature of the IMPRESS system is clearly discernible from output only and the preservation of this flexibility was felt to be necessary.

Tutorial Pedagogy: The interactive nature of IMPRESS has two tutorial thrusts: the instant turnaround on questions to a data base and several adjunct modules which teach the logic behind this approach to survey analysis. Instant turnaround on questions to a data base was not very readily transportable and the compromise represented by POISSON will be treated in detail in the section below entitled "How POISSON Looks." The adjunct tutorial modules are extremely important to transport since they provide another significant resource for customized teaching across a wide spectrum of student backgrounds and learning approaches. Two programs, 2x2x2 and WORDS, were rebuilt to augment POISSON.

Student Feedback: To this writer, tutorial or interactive educational systems suffer generally from the absence of student feedback as a result of usage of a system or even in the design of the system. IMPRESS is not guilty of this omission and the interactive logic and format reflect the heavy student involvement in the evolution of the system. POISSON was built so that the output facing the user was as nearly identical to that of IMPRESS as possible in an attempt to transport this actively sought and received student feedback. Again, it must be noted that the output of student runs and not programs was the only need for transporting the package.

A LAYMAN'S VIEW OF SYSTEMS DIFFERENCES AT THE PARENT AND DAUGHTER SITES

The Honeywell Interactive System (HIS) at use at Dartmouth supports time-sharing almost exclusively. This means that the vast majority of computer runs are submitted through a keyboard with the answers to requests being delivered in a few seconds so that the user apparently is constantly in communication with the computer. The IBM 370 Model 135 (and our previous machine the IBM 360 Model 75) support a time sharing system called the Conversational Programming System (CPS) but this system is not the major support feature as it is in the HIS system. "Batch" computer runs with turnaround time averaging 30-60 minutes take up 75 percent of this machine. In general therefore, the real power of the IBM systems is in "batch" usage as contrasted with the "interactive" usage in the HIS system.

What does this mean for the user? CPS users do not have the overall interactive capacity available to HIS users. CPS not only has a limitation on how many people can get on the interactive system at one time (about 20 as compared with about 90 on the Dartmouth HIS system) but the size of software packages in CPS is also limited compared to those possible in HIS.

On the other hand, the fast core of the TUCC IBM system is significantly greater than that available on the Dartmouth computer and the batch system consequently has far more capacity for handling large data bases and programs. IMPRESS is a tightly overlaid system using a large number of external files. The batch system at TUCC does not suffer from this limitation because of its use of fast core.

The initial problem in designing POISSON was to preserve the interactive nature of IMPRESS as far as possible but to maximize the use of the batch power of the IBM system. The design features decided upon basically set up the interactive system to guide the user toward questioning a data base but restricted the actual calculations to the batch system. Not the least of the considerations leading to this feature was the relatively higher cost of interactive computing versus batch computing, the former being on the average 2-3 times more expensive than the latter.

Since a batch system was necessary to design, the BASIC language could not be used for this system and PL/I was selected as the language of POISSON. This decision necessitated the total rebuilding of IMPRESS since both the language and the machine differences made even the transporting of algorithms impractical and really impossible.

Coupled with a batch system is an interactive system preserving the tutorial power of IMPRESS. This system was also rewritten in PL/I since this language is more powerful than the version of BASIC supported at TUCC. This decision was also required to facilitate the communication of the interactive system with the batch system.

Complicating any possibility of direct translation of IMPRESS programs into POISSON are the fundamental machine differences in the two systems. The word size of each machine is different - an outcome of machine design. The staff at Dartmouth actually shuddered when the possibility of algorithm transfer was suggested and this reaction was repeated by the NCRS staff upon looking at the IMPRESS programs.

HOW POISSON LOOKS

POISSON does look almost identical to IMPRESS when only output is seen. However, POISSON is actually four independent systems:

POISSON: Interactive Tutorial
POISSON: Full-Blown "BATCH" System
ADDER: Data Base Conversion System
2x2x2 & WORDS: Adjunct Tutorial Units

POISSON and POISSON together parallel the IMPRESS system and are a combination of the limited interactive CPS system and the powerful batch system at TUCC.

POISSON alone looks almost identical to the IMPRESS system but it actually operates on a very limited file of each data base. This interactive version guides the user with the same branching as does IMPRESS so that the EXPLAIN and DETAIL features for the advice are available. POISSON also provides all of the information possible on available data sets and the variables in each data set (along with the descriptions of the variables and their categories). The user is guided through the selection of grouping options for each variable (even the process of making his own categories from the raw data) and through the selection of statistical options almost exactly as IMPRESS does by means of the limited file built specifically for each data set.

What POISSON does not do is to follow up by answering the question posed to the data base - it merely outputs the properly formulated question for a run into the batch system, POISSON. POISSON does the work and really can be used independently or even without ever going through

POISSON. Once a user has gone through POISSON one, two, or three times, he can directly use POISSON without the tedium of the interactive-tutorial process and can ask an unlimited number of questions to each data base simultaneously.

The compromise represented by POISSON/POISSON has advantages and disadvantages. Optimization of limited terminal availability for interactive computing becomes a reality for the small college users with less than a few terminals and too many users trying to get an hour or so at these terminals. The lowering of cost is also an advantage of going batch once the POISSON interactive system provides the training for this. Dartmouth student feedback [12] has indicated that IMPRESS "experts" learned to by-pass the tutorial device in order to get into the "batch" system directly. This was a reaction against the tutorial tedium.

The ever-ringing debate over the educational advantage of almost immediate answers to a first hypothesis [13] points out a disadvantage of POISSON. The batch turnaround is problematic but the trade-off with cost and terminal availability makes POISSON more than viable. Output in the current version of POISSON is a production of card formats for input data to POISSON so that students currently are forced to keypunch after running POISSON—a two-step process. Direct feed from POISSON into POISSON was not possible with the version of CPS used up to January 1, 1972, but the introduction of TSO (Time Sharing Option) at this time will make the direct feed possible and the second stage will be eliminated.

ADDEM was built to allow a new data set to be added to POISSON. The automation of this process has opened the possibility of expansion so that four additional data sets to PRES68 and ETHNO are currently being adapted.

WORDS and 2x2x2 are tutorials dealing with analysis of the user's own data for 2x2x2 contingency tables. These programs not only teach gamma analysis but also allow the analysis of available surveys from the popular media and formal publications, given the ability to group variables as standard dichotomies. Of course, these programs run in conversational PL/I at TUCC.

LIMITATIONS OF POISSON

Of the two main branches of IMPRESS, discrete and continuous analysis, only the discrete path was made possible in the first version of POISSON. This limitation is only a result of convenience since the addition of the second branch was decided to be developed after the discrete branch became reliable and relatively debugged. Several other options were set aside for a later version for a realistic beginning. The absence of a continuous branch makes regression analysis and modeling as yet impossible but Meyers of IMPRESS has indicated this branch to be in relatively infrequent use.

IMPRESS allows user defined files to be built in order to save intermediate results which subsequently would save repetition by the user. This file capacity is not necessary in a batch system and does not exist in the POISSON interactive system since no results are produced by this system. Several IMPRESS options related to this file saving are not in the POISSON system.

Table 1 lists the commands available in both systems.

TABLE 1
Commands (Discrete Mode)

<u>POISSON</u>	<u>IMPRESS</u>
XTAB	XTAB
MARG	MARG
	ASSIST
	ITEM
	SCALE
	EFFECT
	IDEA
	STD

The ASSIST command is a tutorial considered unnecessary in the first version of POISSON. ITEM (item analysis), SCALE (Guttman Scaling routine), EFFECT (effect parameters), and STD (test factor standardization) were left to a second version. IDEA (computation of significant associations) was left out due to the availability of WORDS and 2x2x2 considered to effect the same results. Since MARG (marginals) and XTAB (cross-tabulation) are the work-horses of the IMPRESS system, the first version of POISSON was limited to these commands.

Table 2 lists the "statistical options" available under cross-tabulation for both POISSON and IMPRESS.

TABLE 2

"Statistical" Options for XTAB

POISSON	IMPRESS	DESCRIPTION
SELECT	SELECT	Partitioning by inclusion
REJECT	REJECT	Partitioning by exclusion
o/o A	o/o A	Table of percentages across
o/o D	o/o D	Table of percentages down
o/o TAB	o/o TAB	Table percentages on total
CHI	CHI	Chi Square
DELTA	DELTA	Table of observed minus expected frequency
EXP	EXP	Table of expected frequencies
FREQ	FREQ	Frequency table
GAMMA	GAMMA	Goodman and Kruskal's Gamma
NOTAB	NOTAB	Suppresses all table output
	Q	Yule's Q for dichotomous data
	Q*	Yule's Q with brief interpretation
	C	Pearson's C
	D	Somer's D
	LAMBDA	Goodman and Kruskal's Lambda
	NONTAU	Goodman and Kruskal's Tau
	ORDTAU	Kendall's Tau-ordinal data
	PHI	Phi
	T	Tschuprow's T
	V	Cramer's V
	Y	Yule's Y

Absent options in POISSON are a result of convenience and can be added later.

ACCEPTANCE OF POISSON AND ITS FUTURE

POISSON has been used in some way in 10 colleges in North Carolina after only three months of existence. Its quick acceptance was a result of workshop exposure and the preparation involving Dartmouth personnel. It will take at least a year more before this acceptance can be analyzed with regard to the package's reliability, to its preference over other systems, to the absence (to the user) of alternate systems, or to its validity. One inherent advantage of IMPRESS has been the ease by which a user can contact a data base without programming experience or the need to digest three levels of manuals required by systems exemplified by SPSS. POISSON should add criteria to test acceptance as a result of this advantage.

The continuous branch is projected for implementation in 1972. Other additions will be made ad hoc. Four data bases are being prepared for addition to the system. Future development will depend on the continuous acceptance of POISSON in a predominantly batch environment already powerful in SPSS and large statistical systems.

The current version of POISSON has been exported to the University of Iowa and its network. The batch system requires 167K of core and is in PL/I. NCECS is eager to provide tapes containing the entire system (batch and interactive) as well as the two inverted data bases, PRES68 and ETHNO to prospective centers having these minimum requirements.

REFERENCES

1. Interdisciplinary Machine Processing for Research and Education in the Social Sciences. Edmund D. Meyers, "An Introduction to Project IMPRESS," Time-Sharing Colloquium, Kiewit Computation Center, Dartmouth College, February 19, 1970.
2. Of hundreds of visitors to Dartmouth who were interested in moving IMPRESS, the author represents the only visitor to actually implement the system at home. Private communication from Edmund Meyers, Director of Project IMPRESS.
3. J. A. Davis, "Using the IMPRESS System to Teach Sociology," Proceedings of the Second Conference on Computers in the Undergraduate Curricula, pp. 382-388, Dartmouth College, June 23-25, 1971.

4. E. D. Meyers, "We Don't Know What We're Doing," *ibid*, pp. 159-170.
5. R. Anderson, "A Survey of Application Software for Social Data Analysis Instruction," *ibid*, pp. 135-141.
6. E. D. Meyers, "IMPRESS and Undergraduate Education in the Social Sciences," Proceedings of the First Conference on Computers in the Undergraduate Curricula, pp. 8.23-8.29, The University of Iowa, June, 16-18, 1970.
7. No attempt is made by the author to total the grants made by several foundations nor the contributions by Dartmouth to IMPRESS. However, a million dollars is by no means an exaggeration.
8. J. A. Davis, "Elementary Survey Analysis," Prentice-Hall (Methods of Social Science Series), Englewood Cliffs, New Jersey, 1971.
9. IMPRESS Staff, "The IMPRESS Primer," Second Edition, published by Project IMPRESS, Dartmouth College, 1971.
10. M. Mozley and J. Deak, "The POISSON Primer," published by NCECS, P. O. Box 12175, Research Triangle Park, North Carolina, 1972.
11. Inter-University Consortium for Political Research.
12. Town Meeting in Social Sciences, Second Conference on Computers in Undergraduate Curricula, Dartmouth College, June 25, 1971.
13. F. H. Sim, L. S. Rosen, and G. D. Kreider, "Instant Turnaround and Conversational Computing as Instructional Tools in the Social Sciences," Proceedings of the Second Conference on Computers in the Undergraduate Curricula, pp. 142-151, Dartmouth College, June 23-25, 1971.

COMPUTER APPLICATIONS FOR SOCIAL SCIENTISTS

Thomas R. Kerahner
Union College
Schenectady, New York 12308
Telephone: (518) 386-8751

There is a growing interest by college faculty members nationally in computer applications, both in teaching and in research areas. This conference and others will provide reports, in many disciplines, concerning innovations in computer use to achieve many educational objectives. Some new courses built around computers will be added to the undergraduate curriculum. Many other courses will integrate simulation models, hypothesis testing, and the like into exciting courses; these computer-oriented applications will measurably increase interest, relevance, and academic content. The growing availability of time sharing terminals as well as data processing facilities insures that these approaches will be open, at least potentially, to ever-growing numbers of faculty and students in the years to come.

A critical problem is how to get additional colleagues interested in and knowledgeable about quantitative, computer-oriented classroom projects. Such computer projects add interest to classes, as they often enable instructors to deal with more realistic problems. The problem turned on four parallel needs: 1) how to stimulate undergraduate faculty to use computer-oriented applications in their teaching and classroom assignments; 2) how to inform them of the many ongoing models, simulations, and related computer applications currently available commercially; 3) how to suggest computer-oriented projects based entirely upon on-campus materials they could assign to their classes; and 4) how to provide college faculty with the necessary quantitative background so that they would be comfortable in their role as teachers when they introduced this material.

How can these needs be met? Several social scientists from Union College and Howard University, together with the Center for International Programs of the New York State Department of Education(1), asked themselves if they could design an experimental pilot program that would meet the needs of college faculty members with considerable interest but little or no training in quantitative analysis, let alone computer applications. The answer, not unexpectedly, was yes. The result was an intensive summer institute for experienced college faculty, sponsored by the New York State Education Department and held at Union College in the summer of 1969; the program was generally felt to be so successful that it was repeated in 1970.

The purpose of this paper is to offer a preliminary report on the achievements of these two institutes. Short-term summer institutes are widely recognized as a primary vehicle for disseminating new developments—both teaching and methodology—for experienced college faculty(2). In the hope that these may be elements of this program worthy of adoption elsewhere, this paper will highlight both the gains and the special problems these seminars have experienced.

Educational Objectives

It is not expected that a short-term institute could fully substitute for one or more years of graduate training in quantitative methods or computer applications. Few faculty, however, have the opportunity to return to graduate institutions for such training. An intensive, tightly structured summer institute can present the basic theories of quantitative methods. It can highlight and illustrate a variety of their uses, particularly as they are keyed to classroom instruction. It seemed reasonable to expect that a successful institute would have a substantial impact upon both the content and the quality of undergraduate, and even graduate, teaching. Institute participants would be able to offer new courses and certainly widen the quantitative, computer-oriented applications in current offerings. It was also hoped that institutes of this kind would have a significant impact on the undergraduate programs of each participants' college, since a faculty member attending the institute can serve as a major stimulus and a source of assistance to colleagues as well as students.

We also expected that institutes of this type could have a significant payoff in terms of research. The participating faculty have a greatly increased understanding of the quantitative literature in their field. They would also, through assigned problems and projects, have acquired some competence in using quantitative methodology for their own research projects. (Many participants brought their own data to the institute, and over the course of the summer learned several new ways to analyze and interpret it.) It should be emphasized that a dozen or more faculty, over two summers, had developed real quantitative research skills by the end of the program. Sabbaticals or research grants can support additional formal study for some faculty, and virtually every faculty participant had sufficient background to continue and extend his quantitative skills himself.

The Program

While the opportunities for computer applications in undergraduate programs have grown rapidly, a virtual prerequisite for widespread computer use is a solid grounding in statistics and quantitative analysis. The necessary mathematical background and techniques, however, are not widely understood. Many social scientists, whether economists, political scientists, sociologists, or historians, received little or no training in statistics or quantitative methods, let alone computer applications, during their graduate training. It is difficult for the most conscientious faculty member to master this theory on his own. Yet this quantitative methodology lies at the heart of much of the influential work in the social sciences; it has come to be widely represented in the published literature. We recognized the need to develop a program where experienced faculty could acquire the quantitative tools that must accompany computer-oriented applications and innovations.

What were the key elements that must be included in such a program? Put another way, how much mathematical background was required before participants could deal effectively with the material and its computer applications? The program coverage was also limited by the short time period. A too-ambitious seminar, no matter how intensive or how carefully structured the presentation, risks overwhelming the students, so that they cannot work with much of the material 'covered.' We questioned how much computer programming instruction the participants should be given, and we also debated how much quantitative theory, as opposed to applications keyed to illustrative data sets, should be emphasized.

Based upon our experiences during the two summers, it seemed useful to meet twice daily, for 1 1/2 to 2 hours each time, for regular 'classes,' all held in the mornings. These were typically informal lecture meetings, often with frequent questioning. Thirty-minute discussion periods followed the 'classes,' providing considerable opportunity for give-and-take. Afternoons and evenings were largely free for computer work and reading, though we tried to have one 'methods' seminar a week.

A detailed discussion of the topics covered during the seminar is not central to this paper(3). The major areas covered by the program are highlighted below, however, as an indication of representative topics.

In terms of specific coverage, the institute began by examining the foundations of quantitative social science and the role of multivariate causal analysis. From the beginning we discussed the concepts of significance, association, and spuriousness; and the larger question of the difficulties of distinguishing causal from spurious statistical associations received emphasis throughout the seminar. A full presentation of statistical theory laid the groundwork for much of this material. The topics covered included descriptive statistics, stressing data presentation (histograms, cumulative distributions) and statistical characteristics, highlighting standard deviations, the normal curve, Chebychev's inequality, and the like.

The general linear model was studied in some detail. Since regression procedures lie at the heart of much of the quantitative work, the institute fully discussed the assumptions underlying linear regression and then examined analysis of variance and covariance. We applied these techniques, using prepared computer programs, to several data sets. This work was followed by multiple equation dependence analysis, including model building; this led to simultaneous equations, identification problems, causal ordering, and estimation procedures. Factor and discriminant analysis were presented, as were some nonparametric statistics. Many other topics were considered for one or two days.

As to the specific role of the computer, we found that it was highly instructive to provide only a brief discussion of Fortran programming, with much of this concentrated during the first part of the summer. Participants learned to write their own programs, which had the special virtue of facilitating their subsequent use of time-sharing consoles. Generally, however, the institute relied heavily upon sophisticated prepared programs both for the assigned exercises and for the applications to each participant's particular field. Instead, we devoted most of this time to informing the participants about particular computer applications, emphasizing projects they could adopt to their classroom need; and showing them how to prepare instructions and data for the many 'canned' programs.

In sum, the twin concerns of the seminar were constantly stressed. First, we tried to show how a wide variety of quantitative techniques could be used, with computers to answer questions in the social sciences. Second, we attempted to provide enough of a quantitative background so that experienced faculty would feel comfortable in their role as teachers when they introduced this methodology to their classes.

The Participants

The institute was intended primarily for experienced faculty members of colleges and universities who had generally completed their doctorates without significant training in quantitative methods or computers. Participants were selected from several social science disciplines, including economics, political science, sociology, and history. Since such of the methodology was common to the four fields, no one discipline was allowed to dominate the enrollment in either year. Because of the large number of applicants, only about one candidate in five could be offered a place in the institutes. The selection criteria included the customary academic and professional qualifications, together with indications of strong interest on the part of potential participants. We were particularly interested in statements by the applicants indicating how participation in the program would benefit both their colleges and their personal development. To stress our concern that the program have a substantial impact upon the curriculum at the applicant's institution, we requested that the Dean of the Faculty provide a letter assessing both the candidate and the probable impact of his participation upon his department's curriculum.

The Results

There are two primary criteria that were used to judge the effectiveness of the summer institutes. One was the joint evaluation of staff and participants, at the end of the summer, in terms of the effectiveness of the program in meeting its objectives. Were the faculty participants able to satisfactorily understand and absorb the quantitative tools, become familiar with computer usage and applications, and discover several ways in which this methodology could be used to enrich or revamp their undergraduate courses? Perhaps more important, however, is to assess the effectiveness of this type of program in terms of its carryover impact upon the teaching and professional development of the participants. What actual, concrete applications have emanated from or been stimulated by successful completion of this program?

The results from regular assignments, projects, both formal and informal discussion between participants and staff strongly reinforced our feeling that a tightly-integrated intensive institute program could accomplish our objectives. Most participants did possess a broad understanding of quantitative techniques, and they displayed their understanding by completing several computer exercises using a variety of data sets. In classes and seminar discussions we felt that the participants became familiar with a broad sampling of the literature. We also felt that, both through the formal program and by learning from each other, faculty participants would be aware of the methodological strengths and weaknesses of alternative techniques. In sum, the institute's staff generally agreed that the objectives of the program were realistic, and that most participants lived up to our expectations(4).

Evidence on educational carryover is more inferential. Nearly a dozen participants reported that they were able to (and felt comfortable in) introduce new, quantitatively oriented courses in their disciplines. Regression studies were widely instituted. Several faculty, particularly political science, successfully introduced simulation exercises and behaviorist 'games.' In terms of professional impact, at least five participants have written articles that rely upon their newly-acquired mathematical training.

In short, there is considerable evidence that programs of this type can be successful. Conferences such as this often focus on disseminating new teaching developments and computer innovations to faculty who are already familiar with the general approaches. We also see the need, however, to attract and train many additional social scientists so that they, and their courses, can benefit from these 'realistic' applications. Our experience at Union College leads us to call for the sponsorship of many more opportunities for quantitative training for experienced faculty, whether along the model described above or in other formats.

FOOTNOTES

1. The formal title of the State Education Department is The University of the State of New York. As such, it should not be confused with the State University of New York (SUNY) and its 70-some campuses, though it does have the administrative responsibility for the SUNY system as well.
2. Thomas E. O'Connell, Community Colleges: A President's View (Urbana, Ill., The University of Illinois Press, 1968), pp. 24ff.
3. The author would welcome corresponding with interested readers on both topics, readings, and problem sets used in this institute.
4. To our knowledge this program was the only summer institute sponsored by the N. Y. State Education Department that was ever renewed for a second year.

TOWARD THE OPTIMAL USE OF COMPUTER SIMULATIONS IN TEACHING
SCIENTIFIC RESEARCH STRATEGY

John B. Thurmond and Arthur O. Croser
University of Louisville
Louisville, Kentucky 40208
Telephone: (502) 636-6107

In this paper we will describe the use of computer simulations of advanced problems as part of the undergraduate experimental psychology course at the University of Louisville. The problems were conducted in a way similar to the DATACALL game simulations reported last year at the Dartmouth Conference by Dr. Richard Johnson[1] of Earlham College, and Dr. Dana B. Raia of the University of Michigan[2]. Some of the computer simulations developed by teaching fellows at the University of Michigan were used to teach the students the research strategy they needed in order to tackle the more advanced problems used in our course. Psychology 311 at the University of Louisville is a three-hour course consisting of four sections with a maximum of 20 students per section. Each of the sections is taught by a different faculty member with the aid of his own graduate teaching assistant. The course is required for all undergraduate psychology majors, and they typically take the course during their second or third year after completing six hours of introductory psychology. At present, the course can be taken either before, after, or concurrently with statistics.

Prior to introducing the computer simulations this past year, the undergraduate experimental psychology course was taught according to the traditional model of the elementary psychology laboratory. The student was required to read the area's scientific subject matter rather broadly--material which did not stimulate the student to pose questions or seek answers, and most of which was only peripherally related to the classroom activities and laboratory exercises. The student conducted representative demonstration experiments that could be performed within the two-hour class period. He practiced the prescribed laboratory techniques, wrote up the results of the exercises in a prescribed form, and received a grade based on these reports and tests aimed at assessing the degree to which the textbook material had been committed to memory. Classroom activities consisted of lectures that involved the students little, if at all, in the scientific thinking endeavors characteristic of the subject matter area. Finally, the laboratory exercises, although perhaps discussed in class, did not follow as logically deduced experiments that were meaningfully related to the students' reading and classroom activities.

Dr. John Thurmond, one of the course's regular instructors, became increasingly disgruntled with this traditional approach to teaching students, and about two years ago decided that the only way that students could learn to think like scientists was by acting like them. It was realized, however, that the student could not be afforded the opportunity for doing this within the constraints of a three-hour laboratory course unless the process was made much more efficient by compressing the time dimension with simulated experiments. The use of computer simulations as a replacement for the time-consuming steps in the traditional teaching approach permitted the development of a new approach that focuses on activities that are fundamental to the solution of real research problems.

The researcher usually has a considerable amount of information about a research problem before he begins hypothesizing the outcome of experiments. However, he does not know what aspects of his knowledge about the phenomenon are relevant to the particular questions he has formulated, and it is not until he has discovered the important relations between the variables involved that he begins to distinguish the relevant from the irrelevant. Thus, much of the information needed to deduce the experimental outcomes is generally known for any scientific research problem, but it is "hidden" by a lack of perspective concerning the nature of the problem. In order to get the students involved in the kind of thinking activities that a researcher uses in discovering new relations within a knowledgeable framework, they were given a carefully prepared description of the research problem to be investigated. This description, such like a greatly expanded "scenario" that was designed to accompany the DATACALL game (1), gives the student some background on the problem and tells him about the research that has been conducted in relation to it. The relations between some of the variables studied in investigations of the phenomenon described are stated quite clearly in terms of experimental results. The really important relations are suggested repeatedly in the description, but they never appear conspicuously as ones that experimenters in the past have been concerned with directly. These important variables are "hidden" in the sense that they are confounded with the variables the past experimenters have manipulated.

It was planned that the students would apply the knowledge they gained from the problem description to the investigation of the phenomenon themselves by means of experiments simulated with the Psychology Department's PDP-9 computer. This would put them in the position of an actual researcher who must consider carefully and analytically the information at hand, using it to deduce fruitful hypotheses. Based on Thurmond's preparation of these materials, a fellow

faculty member, Arthur Croner, was about to begin work on the program for simulating the problems when he attended the Dartmouth Conference and met Drs. Johnson and Main. We realized quickly that the DATACALL game and computer simulations they were using would be ideal for introducing our students to the more advanced problems we planned to use. Using their programs and modifying them to be run on our PDP-9, the model dealing with schizophrenia and the one dealing with imprinting (both developed at the University of Michigan) were used in a DATACALL format near the beginning of our course in the fall of 1971. The simulations were handled by means of a teletype in the classroom connected to the computer some distance away. This approach proved to be remarkably successful in teaching the students effective strategies for conducting experiments and interpreting results in connection with their investigations of the more advanced problems later in the course.

The computer simulations employed during the first part of the course also provided an appropriate and useful format for simulating the more complex problems. In order to permit the student to discover the really important variables hidden in his problem description, these hidden variables were not listed and described for the student in relation to the computer simulation. The variables shown clearly in the problem description as having been studied by previous investigators were listed for selection in the simulation just as they were in the University of Michigan simulations (2), but there was in addition an "I" variable. In order to test his hypotheses about the unknown "I" variable, the student specified the RANGE of the variable he wished to investigate, stated the RANGE over which he would permit the variable to take on values, and then entered the PARTICULAR VALUE for which he wanted the computer to generate results. Along with the name of the variable, the range, and the value, the student had to enter his variable's CODE NUMBER which he obtained from the instructor. The particular code given to the student was based on his description of the experiment he wanted to conduct, and the computer put in the appropriate effects depending on the numerical value of the code, and the range and value specified.

Before the introduction of these more advanced problems with computer simulations containing unspecified variables, all variables were specified (in the earlier problems), and the student's task was to determine which variables had effects and how big they were. In the more advanced problems, the student's task was to predict the effects of the specified variables on the basis of the information in his problem description, and to formulate hypotheses concerning the effects of additional variables.

Guidelines for Simulation Problems

The problem selected for description and simulation must be one that has been investigated systematically enough to have produced some convincing hypotheses concerning the underlying causes of the phenomenon in question. That is, the students must be given a background leading up to this problem and enough information to ask questions and develop hypotheses. The instructor must be given the knowledge of how the mechanism works (or relevant concepts) and what crucial studies led to its elucidation. The concepts elucidating the phenomenon under study must be relatively simple and straightforward. The subject should be interesting to the undergraduate student--i.e., if not "relevant" to those aspects of human behavior involved in today's problems, it should at least have some important implications for human behavior generally. Not only should the concepts underlying the phenomenon be kept to an absolute minimum, the logic leading to experiments that will produce the concepts should be crystal clear. The problems used in a course should be at different levels of difficulty, with the easier ones tackled first. Finally, the problem descriptions and computer simulations should serve as the vehicle for introducing the student to current knowledge of some of the important concepts of his discipline. In the undergraduate psychology course at the University of Louisville, the problems developed led the students to consider mechanisms of emotion and motivation, short-term and long-term memory, and selective attention. These criteria are characteristics of the simulation problems developed at the University of Michigan, and this, no doubt, contributed to the success of the two used in the first part of our course.

Description of Advanced Problems Developed for Course

Two problem descriptions and computer simulation programs were developed and used in the course. A brief description of them is given below. (1) FAT RATS AND PLUMP PEOPLE. Why do fat subjects, rat or human, eat more than normals when food is easy to get at, but less than normals when the situation makes it a little harder to get at the food? Actually, there are a number of interesting facts about obese humans and rats which show coincidence of behavior. It is not surprising that obese humans and rats eat more "good" food than normals. But it is surprising that the obese, both human and rat, eat less than normals if the food is not too appetizing even if it is the only food available. Among the factors that can be related to eating behavior and obesity are the effects of taste, preloading with food (i.e., hunger), food visibility, prior taste of food, the prominence of food cues, and frequency of eating. The extensive literature concerned with eating behavior of rats with brain lesions when related to the human data is

beginning to suggest that the obesity of rats and men may have a common physiological locus in the ventromedial hypothalamus. (2) THE COCKTAIL PARTY PROBLEM. How do we recognize what one person is saying when others are speaking at the same time? A number of possible explanations for this phenomenon might be explored, among them being those related to lip-reading, gestures, and the like, those that relate to differences in speaking voices such as male and female, loudness differences, pitches, accents, location relative to the listener, and differences related to transitional probabilities in the subject matter and syntax of the language. Investigations of the phenomenon in recent years have been conducted within the framework of theoretical formulations emphasizing mechanisms underlying selective attention and memory.

Programming Considerations

The entire set of programs used were written in FORTRAN IV and were based upon the form described by Main. (2). Considerable re-programming was necessary to allow the programs, run originally on an IBM 360/67 with 1.5 million bytes of core, to be run in a machine with 8K words of core. The effect of this was to make the programs even more modular than before, but depending heavily upon the overlaying capability provided by the manufacturer.

The form of the program was identical: a main program called in each of four links in succession, with ten subroutines distributed among these links. In order to convert a program from one experimental model to another, it was necessary to modify the subroutine that read in parameters and the subroutine that actually contained the model. In addition, some shifting was usually needed to compress the program as much as possible.

Presently we have five programs working that can be made available upon request. Two more programs are in the process of being written, and all seven will be converted to the BASIC language this spring, for use on a time-shared computer which has become available.

Evaluation of Approach

The advantages of using computer simulations to stimulate student interest and participation in designing research was reported at the Dartmouth conference last year by Drs. Johnson and Main. Our use of a similar approach incorporating more advanced problems and simulations has substantiated and extended their findings. The students accepted the computer simulations as validly reflecting the phenomenon under investigation. In fact, they tended to view the computer as "omnipotent" and exhibited little patience with any constraints imposed on the experiments they wanted to conduct. By the end of the course, the students typically showed a preference for designing their own experiments, which were in many cases more sophisticated than those permitted by the computer simulation. For example, they wanted to make inferences that could not be made on the basis of measuring the amount of food eaten by humans and rats under various experimental conditions.

The advanced problems forced the students to think a great deal more about what they were doing in the computer simulations and why they were doing it. Their ability to completely ignore the information they had been given about the variables in the simulation was astonishing. They resisted having to engage in any serious thinking about the variables involved, and wanted to determine effects simply by manipulating the variables in the simulation. Thus, the more advanced simulations with more complete problem descriptions and unspecified variables forced the students away from manipulating variables and more toward thinking like a researcher.

From the instructors' point of view, the computer simulations provided the stimulation and facilitation needed to improve their teaching effectiveness. This became very obvious early in the semester on the few occasions when the computer was "down", and a class discussion was substituted for the computer simulation. These discussions centered on what we would do if the computer were available for the simulation, and this led the students to a much more careful consideration of their research designs. The use of the simulations also precipitated considerable discussion among the instructors and teaching assistants concerning what they were teaching the students and how effectively they were doing it. Some of the advantages for improving teaching that the approach offered in our course were outstanding:

1. The students get a great deal of practice in deciding on research designs and drawing conclusions using statistical techniques.
2. The computer, coupled with problem descriptions, permits the students to simulate many experiments in order to discover new knowledge that they seek.
3. Deficiencies in the student's grasp of the essentials of statistical inference, effective research strategy, and knowledge of relevant material are immediately apparent. Thus, the computer simulations provide an on-going diagnostic tool permitting the instructor to clear up the student's conceptual problems as they arise.

4. The instructor's effectiveness in clearing up the student's conceptual deficiencies are immediately apparent - i.e., if the student does not understand what he is doing, he does not approach the computer simulations intelligently.
5. The approach stimulates attempts to spell out behavioral objectives for each step in the course, and to make a really honest evaluation of what the student is learning.

On the basis of our experience with the advanced problem descriptions and computer simulations, we feel that two questions loom that must be answered in the near future. The first one demands an answer concerning exactly what it is that the students are learning. Certainly the students have no idea. Typically, they have had no exposure to courses that require them to do such thinking, and their idea of effective learning is the commitment to memory of vast numbers of interesting and important facts. Our attempts to evaluate the students with a midterm and final examination met with marginal success at best, indicating that our appreciation and understanding of the learning process engendered by the approach is, at this point, rudimentary. The exams we gave seemed to have a lot of face validity. They defined and discussed a problem area for the student, and included parts that required the student to formulate hypotheses, design experiments, predict and evaluate alternative outcomes, explain positive and negative results, and create entire research programs for investigating the problem with accompanying descriptions explaining his research strategy. We expected that the results of these exams would correlate highly with the student's effectiveness in conducting the computer simulations and participating in class discussions. While the exams did a fair job at the extremes (for the best and worst students), the results for the most part were equivocal. Perhaps the lack of correlation is due to the format of the exams more than to what it was designed to measure - i.e., the use of a written timed exam as opposed to some sort of oral, self-paced procedure. In any case, it is apparent that more work in the future will be necessary to determine how to adequately assess the student's scientific research strategy.

The second question concerns the students' exposure to "live" demonstrations and data collection. The main difference between the approach used in the course described in this paper and the one used by Drs. Johnson and Main [1,2] was the substitution of the advanced problem descriptions and computer simulations for the student's own research project during the latter part of the course. While the use of these advanced problems introduced new levels of thinking and analysis for the students, about two-thirds of the way through the course the necessity and desirability of exposing the students first-hand to the phenomenon under study was felt rather acutely. Hence, prior to conducting the cocktail party problem, the students were exposed to prepared tapes which demonstrated selective listening phenomena, and they participated in a brief but instructive shadowing experiment. This pre-exposure made the cocktail party problem description and simulated experiments such more interesting and meaningful for the students. On the basis of the success of this experience, we plan to make each of the simulations used in the future more meaningful for the students by this sort of exposure to the phenomenon under investigation. By selecting experiments and demonstrations that require a minimum amount of time for data collection, and by permitting the students to analyze their data collected on the computer, the time devoted to this aspect of their instruction should be well spent. Thus, one approach concentrates entirely on research strategy and computer simulations with no data collection during the first part of the course followed by the student's own project with data collection during the second part of the course (1,2). Our approach concentrates on increasingly difficult computer simulations with data collection and demonstrations designed to enhance the student's appreciation and knowledge of each problem. Future work should be carried out to determine the relative merits of these two approaches.

REFERENCES

1. Johnson, R. DATACALL: A Computer-based Simulation Game for Teaching Strategy in Scientific Research. Proceedings of the Conference on Computers in the Undergraduate Curricula, Dartmouth, June, 1971.
2. Main, D. B., & Head, S. Computer Simulations in the Elementary Psychology Laboratory. Proceedings of the Conference on Computers in the Undergraduate Curricula, Dartmouth, June, 1971.

A SERIOUS GAME AS AN INTRODUCTION TO URBAN PLANNING

Aaron H. Konstan, John Bartolomeo, and Judith Johnston
The Lindenwood Colleges
St. Charles, Missouri 63301
Telephone: (314) 723-7152

Introduction

All freshmen students at The Lindenwood Colleges are required during their first year to take what is called the Lindenwood Common Course. The stated purpose of this course is to entice these freshmen, individually and collectively, to consider what the Common Course faculty believes to be the pressing problems of twentieth century man. The course considers such issues as ecology, the urban crisis, changing social values, abortion, etc. During the 1970-71 academic year one of the foci of the Common Course was urban planning. The pedagogical vehicle used to introduce the students to considerations which enter into the planning of a city was an urban planning game called New Town. In this paper the game will be described and the pedagogical results of using New Town to introduce students to urban planning will be discussed.

The Game

New Town was originally developed by Barry Lawson(1) while he was a graduate student at Cornell University and was expanded and modified at Lindenwood. As originally designed, all of the playing and accounting were done by human participants. We found these accounting procedures tedious and time-consuming for the human players; therefore we computerized the game.

Players

The players of the game divide themselves into five teams. Four of the teams represent developers and the fifth represents the public planner or city manager.

Purpose

The purposes of the game are threefold. The first is to develop a city on the available land. The second is for the developers to get the largest possible return on their investment. The third is to develop a city that is worth living in.

As might be expected, the fulfillment of this last purpose is mainly the responsibility of the public planner, but the developers are involved to a lesser degree by means of a "goodness test" built into the game.

Game Board

The development of the city is done on a game board (see Figure 1) which represents the map of the land set aside for the new city. As can be seen in Figure 1, the land has already been divided into blocks, each containing four parcels of land.

The development land contains a lake, a river, and a railroad track, which are intended to be forces in shaping eventual development patterns. All the land is assumed initially to belong to the bank.

Development Units

The developers have a choice of building residences, retail establishments, or industries of varying sizes and densities.

The public planner is responsible for the development of parks, utilities, schools, the town hall, fire stations, sewage plants, refuse disposal plants, health clinics, civic centers, and airports. These units have fixed development costs.

Steps in Play

Each team of developers begins the game with \$1,500,000 in cash. The public planner starts with no money, a \$300,000 debt limit, and a potential income from taxes on the cumulative value

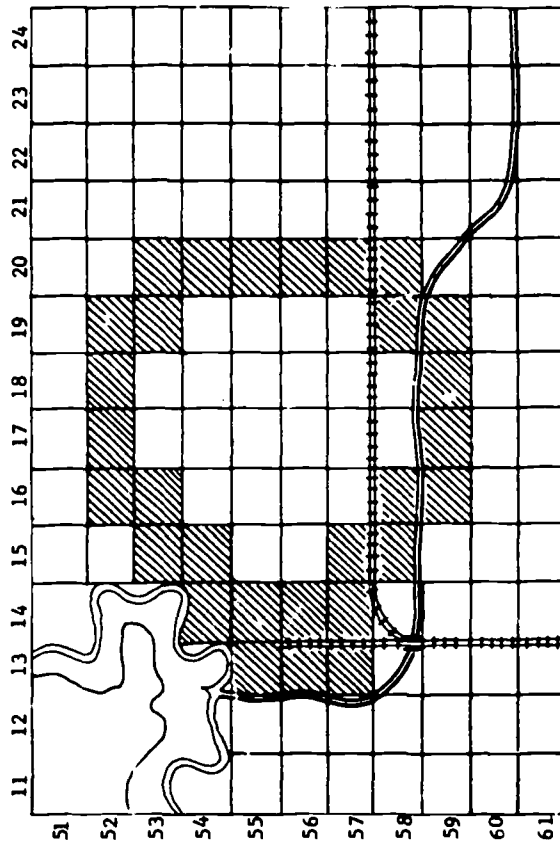


FIGURE 1. Diagram of Playing Board

of the property developed. The normal tax rate is ten percent of the cumulative value. Play occurs in rounds designed to simulate the activities of one year of urban development.

Each round starts with each of the teams purchasing up to five pieces of land by submitting sealed bids. After the land is purchased, the development phase of the game begins with the developers bidding and putting up development units, residential, retail, and industrial. The team representing the public sector can put up any developments consistent with its income, its debt limit, and the fines imposed by the public issue aspect of the game to be described subsequently.

The rules for development were designed to mirror the constraints on development present in a real urban environment. For example, industry built on water and/or rail lines earns higher income since transportation costs are lower; retail units in shopping centers are more profitable; rentals are higher on residences built on lakefront or park-front property, etc.

Developers are allowed to do the things they would be expected to do in real life. They can borrow money, vote to raise or lower taxes, redevelop, raze their developments, bribe city officials, or band together to develop. The planner, on his side, can expropriate property, put up public developments, play the developers against one another by promising to build or not to build, etc., all presumably for the stated end of making the city the best around.

At the end of each round the information on sales, developments, money transfers, loans, rentals and tax rates are coded onto cards and fed into the computer. The computer analyzes the events of the round, does the accounting for the private and public sectors, and produces a report sheet for each team. The accounting is done according to the equations given in Figures 2 and 3, while a sample account sheet is shown in Figure 4.

"Goodness Test"

The "goodness test" is designed to illustrate to the players the problems that arise in developing one piece of property without considering the effects developments have on one another. This is done by having the computer calculate the number of jobs available from the industrial, retail and public developments as well as the number of workers living in the residential developments. If these two numbers are not the same, within certain tolerances, penalties are assigned to the developers in the form of lower business incomes or higher tax rates, depending on the direction of the imbalance.

$$\begin{aligned} \text{Annual Investments} &= \text{Land Costs} + \text{Development Costs} \\ \text{Cumulative Value} &= \text{Sum of Annual Investments} \\ \text{Expenditures} &= \text{Annual Investments} + \text{Rents Paid} + \text{Redevelopment Costs} + \\ &\quad \text{Taxes} + \text{Bank Loan Interest} \\ \text{Money Left Earning Bank Interest (MEBI)} &= \text{Cash-on-Hand from Previous Round} \\ &\quad \text{Expenditures} \\ \text{Annual Income} &= \text{Residential Income} + \text{Retail Income} + \text{Industrial Income} + \\ &\quad .05 * \text{MEBI} + \text{Interest from Private Loans} \\ \text{Net Income} &= \text{Annual Income} - \text{Expenditures} \\ \text{Cash-on-Hand} &= \text{Cash-on-Hand from Previous Round} + \text{Net Income} - \text{Bank Loan} \\ &\quad \text{Outstanding} \\ \text{Rate of Return} &= \frac{\text{Annual Income} - \text{Tax} - \text{Rent} - \text{Redevelopment Costs} - \text{Bank} \\ &\quad \text{Loan Interest}}{\text{Cumulative Value} + \text{Cash-on-Hand}} \end{aligned}$$

FIGURE 2. Accounting Formulas for Private Sector

Public Revenue = Tax Rate * Total Cumulative Value of Four Developers
 + Balance from Last Round + Interest on Private Loans
 + Money from Private Loan (or Gift)

Interest = .05 * Amount of Bank Loan

Expenditures = Land Costs + Development Costs + Operating Costs + Interest
 + Redevelopment Cost + Rent + Penalties

Net Income = Public Revenue - Expenditures

New Bank Loan = Old Bank Loan - Net Income

FIGURE 3. Accounting Formulas for Public Sector

ROUND 1

LABOR FORCE = 34.2 JOBS AVAIL. = 103.0 RATIO LABOR/JOBS = 0.33
 INSUFFICIENT LABOR FORCE. BUSINESS INCOME REDUCTION = 66.7 PERCENT.
 TAX RATE = 10.0 PERCENT

DEVELOPERS REPORTS, ROUND 1

TEAM NO.	RESID. INC.	BUS. INC.	TOT. INC.	TOT. ANN. EXP.	TOT. EXP.	NET INC.	CASH	BANK LOAN	CUM VALUE	RATE OF RETURN
1	32000.	41836.	115836.	600000.	660000.	-544163.	955837.	0.	600000.	3.5
2	0.	0.	75000.	0.	0.	75000.	1575000.	0.	0.	4.7
3	0.	0.	75000.	0.	0.	75000.	1575000.	0.	0.	4.7
4	30000.	47813.	123738.	525000.	577500.	-453561.	1046438.	0.	525000.	4.5

PUBLIC ACCOUNT REPORT, ROUND 1

TAXES COLLECTED	112500.
TOTAL REVENUE	112500.
DEVELOPMENT COSTS	260000.
OPERATING COSTS	52000.
INTEREST ON LOAN	0.
LAND COSTS	10000.
REDEVELOPMENT COSTS	0.
PENALTIES	0.
RENT	0.
TOTAL EXPENSES	322000.
NET INCOME	-209499.
BANK LOAN	210000.
BALANCE IN ACCOUNT	500.
DEBT LIMIT FOR NEXT ROUND	281250.

THE PUBLIC ISSUE FOR THE NEXT ROUND IS FIRE STATION.

FIGURE 4. Sample Accounting Sheet

Public Issue

At the end of each round the computer analyzes the "character" of the community already developed and, on the basis of this analysis, decides which type of public development the community needs most. This development is then designated as the public issue for the next round, and the public planner is fined one-quarter of its development cost for each subsequent round it is not built.

Description of the Use of the Game in the Common Course

The students taking the Common Course were divided into nine groups each containing approximately 25 students. The faculty who plans and teaches the course includes three members from the social sciences, three from the humanities and three from the natural sciences. The academic year is divided into four time periods of about seven weeks in length. By rotating the nine student groups among the nine faculty members it becomes possible during the first three time periods for every group of students to spend one seven week time period being taught by a social scientist, one being taught by a humanist and one being taught by a natural scientist. During the fourth time period the students work on independent projects to be submitted at the end of the year.

The students played New Town one day, usually spending five to six hours on the game, during the time period that their group was being taught by a social scientist.

The actual administration of the game was done by the staff of the Computer Center, two of whom, the director and a student assistant, are authors of this paper. After the game was concluded, one of the computer staff, in concert with the Common Course faculty member responsible for that group of students, would spend approximately one-half hour discussing the results of the game while it was still fresh in the students' minds. Students were encouraged to see relationships between the city they had developed in the game and the characteristics of existing cities. Such matters were discussed as the relation between transportation facilities and development of the city, reasons for high concentrations of industry in certain areas of the city, and interrelationships between the locations of different types of developments. The Common Course faculty member was then able to use the experience with the game and the discussion following play as a springboard for greater in-depth examination of urban planning in the classroom.

During the academic year the game was played nine times, once with each group of students in the Common Course. It should also be noted that each of the social scientists on the Common Course faculty was involved in supervising a student group playing New Town three times during the academic year.

Further Use of New Town

During this same academic year one of the authors, who was a sociologist on the Common Course faculty, was also teaching a course in urban sociology. The students in this course were mainly upper classmen majoring in sociology. It was decided to use New Town as an educational tool in this course. The contrast between our experiences in these two courses will be discussed when our results are described.

The Computer

The computer system of The Lindenwood Colleges, which was used for this game, is an IBM 1130 Model 2B, 8K core memory with one disk drive. It is also equipped with a 1442 Model 6 card reader/punch and an 1132 line printer. The account forms were produced on the line printer using five part paper, so that each of the teams received a copy of the Accounting Sheets at the end of each round.

Role of the Computer

In our judgment the use of the computer makes a crucial difference in the pedagogical value of the game. The emphasis in the computerized game is on the role playing in an urban planning situation rather than on the tedium of doing the arithmetic. (It would probably take most students up to one-half hour to do the accounting for each round. The computer analysis of the same data is essentially instantaneous).

It is also possible to change the complexity of the economic model used in doing the accounting without appreciably altering the time the computer takes to analyze the data produced after each round. For example, it could easily be arranged for part of the game to be played without the "goodness test" and part of the game with this test to illustrate the effect that the test has on the rates of return and the other economic measures of the success of development schemes.

Attitude of the Students

Since the Common Course is taken by the entire freshman class, the students who played New Town were of widely different philosophical and political persuasions. Some began playing the game with positive attitude, full of curiosity about what might happen. Others felt that having to spend a whole day on New Town was a great imposition on their personal freedom.

Despite the different initial attitudes, once the students got involved in the game, they seemed to enjoy the experience. Many wanted to continue playing even after the allotted time was spent. It is our experience that the serious game approach to learning has the a priori advantage of being fun. It seems self-evident that a learning experience which is enjoyable is potentially more effective pedagogically.

New Town as a Learning Experience

The crucial question, of course, is whether the students learn anything from playing the game. A further question is whether the serious game has any advantages as a pedagogical tool over other types of learning techniques.

The authors are fully aware of the work of J. Sharac and D. Russ(2) which attempted to measure quantitatively the learning value of a serious game approach to learning. We have no quantitative statistics to refute their findings that there appeared to be no quantitative advantage to the game approach.

Our qualitative experience, however, supports the contention that the advantages of the serious game are not in teaching about the interplay of concepts when applied to a model of the real world. For example, the meaning of natural depreciation of the value of a development, on one hand, and taxation on cumulative value on the other, can be taught separately in the classroom. However, the effect on the tax base of the city of having depreciation on existing developments and insufficient expansion of the tax base through new development is better demonstrated in a game situation. We believe that Sharac and Russ did not put enough emphasis on learning about interrelationships between basic concepts. When these interrelationships are taken into account the advantages of the serious game become apparent.

Our sample was too small to have control groups and to do extensive testing of the learning accomplished. The above mentioned convictions about the positive value of the game experience come from seeing students with essentially no more background than playing the game five or six hours become able to discuss somewhat knowledgeably the rather complicated intertwining problems facing the urban planner.

Our experiences with New Town have demonstrated that the following set of factors affected the efficacy of New Town as a learning experience.

Attitude of Instructor

The Common Course faculty who supervised the student groups playing New Town during 1970-71 came from three different disciplines within the social sciences: psychology, economics, and sociology. As it turned out, these three instructors had varying degrees of commitment to use of New Town as well as to discussion of urban problems.

One of the authors, who also taught the course in urban sociology which used New Town, was, of course, very interested in using the game to as full an extent as possible. Before playing the game, his students spent time discussing the principles involved in urban planning, and received a preliminary introduction to the game. After the game was played, his groups discussed the results in the classroom.

The other two Common Course faculty were not so committed to the whole New Town endeavor. They had participated with the entire Common Course staff in deciding to use New Town but urban planning problems were not of great personal interest to them. They did not spend extra time before or after their groups played the game in discussing the game or what the whole New Town experience was about, nor did they participate with their students in playing the game. They

depended on the game and the brief discussion which followed at the Computer Center to be the complete learning experience.

Our observations indicate that these varying faculty attitudes were passed on to the students in their respective groups. All the student groups learned something from playing the game, but the learning appeared more extensive among those groups of students who sensed a commitment on the part of their instructor to the issues involved. Even the attention span of the students during the explanation of the game rules was markedly greater for those who knew that their instructor cared about the results of the game.

We believe that the importance of the attitude of the instructor might be overlooked in evaluating any learning technique and especially one as complicated as the serious game. To us, it seems crucial that the game experience be one which the students and the instructor share together or it ceases to be meaningful to the student.

Preparation of Students

The fact that we were using New Town both with freshmen in the very generalized Common Course and with upper classmen in a specialized urban sociology course allowed us to compare these two groups and the results of their using the game.

These two groups differed in a number of ways. The upper classmen, it can be assumed were more sophisticated in their approach to learning as well as more committed to the study of urban planning problems. The students in the sociology course did not come to play the game cold as did most of the freshmen. The sociology class had reviewed the rules of the game and had actually played a kind of warm-up game of New Town before they began playing the game in earnest. The rules of New Town are admittedly complex (the full explanation of the game comprises twenty printed pages), and not having the rules clearly in mind prevents a player from exercising freely all his game options.

The following differences were noted in the two student groups. The freshmen, having less preparation, exhibited greater changes of attitude as a result of play. They were dealing with concepts which, in the main, they had not dealt with previously in the classroom, and about which they knew little. Their discoveries of the interplay between the concepts and values embedded in the game were more striking and had a more dramatic effect on their understanding of the problems of the real city.

The upperclassmen also learned from the game, but the game served mainly to reinforce their understanding of ideas which had already been considered in class. Therefore, their ultimate understanding was deeper and more profound but the change from their pre-game to post-game attitudes was less striking. This more profound understanding of the principles involved allowed these students to play what might be termed a tighter game. That is, in playing their game, the teams of upperclassmen made better use of the economic and political forces present in the simulated developing city. Their moves were taken less at random than were the moves of the teams of freshmen, since the freshmen were less sure about the consequences of each move.

Realism in New Town

There appears to be a lack of agreement among the designers as well as among the users of serious games about the importance of realism in the game rules. To illustrate, in the case of New Town it might be discussed whether the quality of the learning experience is affected by giving each team of developers \$1,500,000 as an initial sum of available capital for beginning their development, instead of \$1,500 or \$150,000,000. Since it is a game, does the amount matter?

The authors believe that in order to use the serious game as a vehicle for teaching about the real world, the context in which the game is played should be as close as possible to the context in which the activity the game is designed to simulate is carried out in the real world. In our case, our purpose was to go directly from playing the game into a meaningful discussion of urban planning problems in the real world. For this to be possible it seems self-evident that an attempt must be made to make the game as realistic as possible.

We cannot pretend that all the aspects of New Town correspond directly to features present in an actual urban planning situation. We are aware of several urban planning and land use games, such as CLUG(3), which are in many ways more realistic than New Town. Compromises must be made in order not to make the game too complicated or too tedious to play. But without a realistic base a simulation model has no *raison d'être* since it is in learning through simulation of real life situations that serious games have a useful role.

Interaction of Basic Concepts

As we indicated earlier, our experience supports the conclusion that the serious game is no better than any other method for teaching about such concepts as tax base, rate of return, development costs, etc. All these ideas could just as well be defined and discussed in a formal classroom setting. However, when it comes to illustrating the interaction of all the economic, political, and social forces affecting the private and public development of an urban area and resulting in existing cities with their various problems, the serious game represents a superior pedagogical technique.

Perhaps an example will make this point more clearly. After playing New Town the students were quite capable of discussing such questions as: Why in a city like St. Louis is a large concentration of industry found between the railroad station and the river? Why do shopping centers spring up? Why in cities are there large concentrations of luxurious rental property beside parks and lakes? How are the locations for parks, schools or other public developments determined? How do private developers and the city government interact to develop a city? To answer any of these questions one must have an understanding of the interaction of the variety of forces that are present in an urban environment. With newer versions of New Town which include ecological constraints, students can be led to consider such additional issues as the problems of building on a flood plain, the building of levees, and the effect of industrial pollution on urban growth.

All these topics could certainly be discussed in a formal classroom setting. However, it would entail a great deal of lecturing to give enough basic information to the students so that they could begin to see the interrelationships involved. Information regarding these matters and many other concepts and interrelationships can be made to fall out naturally from the experience of playing a serious game. And, as was pointed out earlier, the serious game approach is less tedious as well as more fun for all the participants than the formal lecture.

This is not to say that playing such a game as New Town has the magical effect of making all the students experts in urban planning. Rather, the game presents to the students in a more concentrated and more palatable form such of the basic information necessary to begin an in-depth study of the planning problems of a city.

NOTES AND REFERENCES

1. Game is available in both computerized and uncomputerized forms from: Harwell Associates, Box 95, Convent Station, New Jersey, 07961.
2. J. Sharac and D. Russ, Evaluation of Student Learning and Information Utilization in a Computer-Simulated Economic Model. Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, 1971, p. 94.
3. Developed by Alan Feldt of Cornell University and available from: Center for Housing and Environmental Studies, W. Sibley Hall, Cornell University, Ithaca, New York

POLITICAL SIMULATION AND THE MINI-COMPUTER
A CHALLENGE TO THE INDUSTRY

Marshall H. Whithed
Temple University
Philadelphia, Pa. 19122

Summary

This article discusses the uses of computer-assisted political simulation models, and the requirements such simulation models impose upon the computer system which is utilized. The author suggests that the educational goals sought in the use of political simulation exercises are best attained through the utilization of a conversational mode time share computer facility, or, alternatively, a dedicated facility, on a conversational interactive basis, although a number of computer problems are thereby created.

The author's work has not been with mini-computers, but rather, with large computers on a time share basis. In terms of the present interest in mini-computers, the author discusses his experiences in the large-computer time share world, with illustrations provided by means of a number of urban simulation models he and his group have been working with in the past several years. The requirements of the work are discussed in detail, from the user's viewpoint, as a challenge to the mini-computer industry. The suggestion is made that a dedicated mini-computer should be able to accomplish the requisite tasks, and maybe do this on a cost-competitive basis when the true, rather than ideal, performance of large-scale time share computers is taken into account.

Political Simulation

Political simulation may be regarded as an experimental technique through which complex political phenomena such as a political campaign or an international relations crisis involving a series of events and a number of nation-state "players" may be "recreated" under quasi-experimental conditions at the will of the person conducting the simulation. As such, then, simulation techniques have the advantage of controllability; that is, the circumstances may be altered at will, and especially, the timing of the exercise may be altered to suit the convenience of the researcher or of the teacher.

Simulation techniques are not new, and in fact have been applied for a number of years in the field of business management[1]. In political science, simulation techniques have been previously applied to international relations and foreign policy situations. A particular case in point is the Inter-Nation Simulation, which has been utilized extensively at Northwestern University and several other universities. A teaching version of this model is marketed in kit form by Science Research Associates[2]. In the field of American Government and Politics, there is an election game developed by James S. Coleman which has been utilized at the Johns Hopkins University and in the Baltimore high schools. A national political game has been utilized at Kansas State Teachers College under the direction of Dale Garvey, and an American Government game under Robert Alparin at the University of Maryland. A preliminary version of a presidential election simulation prepared by Marvin Weinbaum and Louis Gold, since published by Holt, Rinehart, and Winston, has been utilized by this writer on an experimental basis in Basic American Government classes at Northern Illinois University and Temple University.

Most of these simulation models are intended to teach the participants particular skills or how to perform particular functions, such as being the head of state of a country, devising viable military strategies, or managing a large business. A major inducement to the utilization of simulation techniques in such situations is that the participant can learn from his mistakes without suffering the real-life consequences when mistakes are made while learning. A participant in a business management simulation may, through poorly conceived strategies, bankrupt his company and learn from the experience without actually losing real world money. The miscalculated foreign policy decisions of the head of a simulated state may lead to war and thus provide the appropriate lesson for the decision-maker without incurring the social costs of actual war. In short, simulation exercises are useful in teaching skills when the consequences of error in the real life context are so costly as to effectively prohibit trial and error learning[3].

While these reasons are in themselves significant motivations towards the utilization of simulation techniques, there are other advantages as well. In particular, many have long recognized that the academic classroom is not an adequate setting in which to convey to students an understanding of the complexities, inter-relationships, and dynamics of political phenomena, or most complex social phenomena for that matter. Partially in reaction to this problem, teaching political scientists have for some years been utilizing supplementary teaching

techniques such as political campaign work and in-service training in public administrative agencies. Unfortunately, however, there is a definite limitation on the number of meaningfully instructive positions available to college students. As anyone who has included political campaign participation for his students as part of his course requirements is aware, often students involved in a political campaign are relegated to relatively unproductive activities such as licking stamps, answering telephones, and handing out the candidate's literature. Seldom do they find themselves in positions from which they can observe the processes by which the important campaign decisions are made. And it is a very infrequent occurrence when the student finds himself in a position in the campaign organization where he can actually participate in, and thus experience, the decision-making processes.

Arranging for public administration internship or equivalent experiences for students also presents the problem of a limited number of suitable openings, as well as consuming a great deal of professional time in developing and establishing the openings. And many of these openings require a considerable amount of travel for the students between campus and job.

Given these circumstances simulations offer an attractive alternative which provide not only a sufficiently large number of meaningful instructive positions, but ones in which the participants can learn how and why decisions are made through actually functioning as a decision-maker and by experiencing some of the cross-pressures and consequences of decision-making.

In political science curricula we have been experimenting with several computer-assisted simulation models. Among these is an urban land use game, which we have derived from the Community Land Use Game (CLUG) originally developed by Allan Feldt[4]. We have developed for time-share computer facilities, an international relations simulation model with an especial focus on multi-national business considerations, called Politically Simulated World (PSW)[5], and a computer-assisted version of our Woodbury: A Political Campaign Simulation[6]. From our experiences with these models, we have come to several conclusions regarding the role of computers in political simulation modeling, and particularly with reference to educational pursuits.

Basically, we found it necessary to move to the use of computers in order to handle the mass of data inherent in anything more than a very simplistic effort at modeling the political environment within a reasonable time span and with a reasonable effort input. Even with our earlier efforts at batch-mode processing simulations, such as our Politically Simulated World (PSW) exercise, we found it necessary to resort to a computer to make the necessary calculations in time for the next simulation period. In one of our early runs of the PSW simulation, when we were in fact running two separate games, a computer breakdown forced the simulation directors to do the calculations manually; this task took one of us, who is an accomplished accountant, nearly two full days of tedious effort to complete. This type of work load is rather discouraging in terms of everyday class use, and of course would prohibit utilization of the model in a continuous simulation run of many repetitive cycles.

Besides the procedural problems inherent in a batch-mode processing situation, we have found through extensive experience that batch-mode computer processing often means that we cannot feed the results of analysis to the simulation participants at the most appropriate time in the simulation cycle, but rather, must introduce data at an artificially-delayed time in order to allow for batch input and output procedures and handling. And from an educational point of view, batch-mode processing often precludes rapid and meaningful feedback to the simulation participants. This results in less learning than might be the case if more rapid and comprehensive feedback could be provided at the moment of interest, and also perhaps a loss of participant interest which might have been sustained or even heightened if more rapid feedback could have been provided.

The desire for more timely introduction of data from the previous simulation round into the next cycle, plus the desire to provide more rapid feedback to the simulation participants, have led us to explore the time-share computer mode for teaching simulation work. Using a remote computer terminal, a simulation operator is able to input the data resulting from participant decisions in a particular simulation cycle and to receive back on his terminal within a matter of minutes the new simulation data updated as a consequence of the previous participant decisions.

Educationally-speaking, we find that the time-share approach not only provides much more rapid feedback for the simulation participants, but also makes it possible to program the simulation model so that participants can, using the remote computer terminal, explore the potential effects of various alternate decision strategies open to them. In short, time-share makes possible an interactive process between the simulation model and the participants.

Finally, it is perhaps worth noting that a carefully-constructed educational simulation computer program can, with a well-laid out series of query statements through the remote terminal, make the administration of a complex simulation exercise much easier for the

simulation director. And anybody who has run a session of hand-controlled CLUG or another similar simulation exercise can well appreciate this last.

In order to accomplish these objectives, and especially the last, it is necessary that the computer program "ask" the relevant decision-point questions at the appropriate point in the simulation cycle. Furthermore, these queries must be made in a clear manner easily understandable to the user of the simulation model (rather than being structured according to the computer technician's point of view), and the response patterns should be clearly apparent to the user. In short, we here envision a conversational mode of man-computer interaction via remote computer terminal, and this is only possible through the time-share environment.

In the paper we discuss our experiences in developing a computer-assisted version of the Woodbury Political Campaign Simulation, including an analysis of the educational motivations which led us to perceive the necessity of a time-share environment and conversational approach. Next, we examine an urban land use simulation model, CLUG, in its human-controlled and early-version batch-mode computer-controlled version for the IBM 1180 as developed at Cornell University. In particular, we find that with either the human-control or batch-mode computer approaches, the necessity of emphasizing accounting and preparation of specific format inputs on punchcards distracts greatly from the simulation participants' abilities to concentrate on the substantive aspects of land use development because they must of necessity spend too much of their time on the mechanics of filling out paper forms or punching specific format Hollerith cards. We therefore have developed a time-share version of a modified CLUG model, in which the conversational "abilities" of the time-share approach rectify this problem, and also make the exercise much easier for a simulation director to administer.

Finally, the possibilities of using mini-computers to fulfill the same educational objectives that led us to use time-share facilities are analyzed and discussed.

Urban Electoral Simulation Exercises

In the urban context, a number of urban land use development and urban electoral simulation models have been developed, including the ones we report on here. Some of these models have become quite intricate, and embody the land use developmental, political, social, and economic sectors of the urban polity. Some of these models have been used for teaching and, tentatively, to explore with urban decision-makers some of the outcomes of various alternative strategies of action open to them.

The Woodbury urban mayoralty election simulation is a simulation of urban-oriented political interactions. As played thus far, with an exception to be noted later, it has been a "man simulation," that is to say, human actors play various roles, and the scoring is performed by human umpires. The scenario for the exercise portrays a hypothetical community, is constructed for teaching purposes, and so contains a considerable number of urban problems for the simulation director to explore with his students.

The participants occupy various campaign roles, such as the mayoral candidate's organization, pressure groups, the mass media, and so forth. Their actions are constrained by the parameters contained in the scenario documents, and as the simulation exercise unfolds, new parameters upon their actions are established by the activities of the other role-playing groups.

The 'pay-off' mechanism for the role-playing groups is a two-fold one. The first level of payoff is the simulation election for mayor, for obviously one candidate wins and the other loses. This function is performed by the umpires as described below. The second level of payoff, centering upon the various pressure groups, is generated by the simulation participants themselves in their peer group interactions. Each pressure group formulates a goals list and a strategy to achieve those goals through gaining the assistance of other groups, and ultimately, the support of one or both mayoralty candidates for their position.

Basic to the man-simulation model of the Woodbury simulation is the function of umpire. Woodbury is divided into nine wards (ten wards in a special ten-ward model), and an umpire is assigned to each ward. In our runs of this simulation exercise, the umpires have been graduate students and professors of political science. Probably a group of umpires could also be recruited from knowledgeable townspeople, politicians, and civic groups. The umpires assess the various moves of the roleplaying groups, moving prospective votes from one candidate to another and/or to or from the politically independent column. The roleplayers generate numerous moves during a simulation run--some Woodbury games have generated more than 200 moves to be assessed.

The utilization of human umpires who are particularly knowledgeable about the subject matter to do the scoring and thus control the outcome of the simulation exercise is similar to the use of experts in the field of international relations and diplomacy to judge the direction

of an international relations political simulation exercise, the Political Military Exercise or PME[7].

In our experience, the man-umpire approach to the Woodbury-type urban politics simulation model has a number of advantages, particularly for teaching purposes. But on the other hand, the necessity of numerous ward umpires poses a serious logistical problem as the number of wards in a city to be simulated is increased. Also, the logistical coordination burdens in the man-umpire approach more or less effectively preclude such extension of the model to take into account other variables. And for regular teaching purposes, the necessity of finding nine political scientists for the simpler Woodbury-type model may pose problems at smaller colleges.

For these reasons, we have developed a computer-assisted version, and have utilized the model in the classroom[8]. In the computerized version of the Woodbury model, the various groups in each ward are treated as "Voting Block Reaction Groups." As in the original man simulation model, student role-playing groups initiate moves or political actions. The simulation director codes the verbally-expressed moves into a preestablished coding scheme in the same manner as open-ended survey questionnaire responses are coded into numeric format for computer analysis. When inputted to the computer according to the predetermined coding scheme, these moves are "evaluated" in terms of their specific and cumulative effects upon the VBR Groups of each ward. From the data base stored in the computer program, the reaction weightings for the VBR Groups in each ward are generated on a +9 scale. Vote distribution shifts of potential voters are reflected in the "Gallup Polls" provided to the student participants through the remote computer terminal as indicators of the success or failure of their strategies. The vote distribution shifts are totaled for each Gallup Poll, and at the conclusion of the simulation exercise the computer translates the potential voter data of the Gallup Poll into "Actual Vote" totals for each ward, according to preestablished formulas representing the differential between potential voters who would be tapped in a political survey (Gallup Poll) and the actual voter turnout at the polls (as expressed in the voting behavior model for the simulated city, upon which the computer weightings and formulas are based).

The computer printout provided to the participants during the simulation exercise is designed to provide role-playing teams with detailed information regarding the political effects of their specific political moves. Because of the rapid interaction between the role-playing team participants and the simulation model through interactive conversational mode time-share computer terminals, quick feedback can be provided to the participants. Additionally, role-playing groups can utilize the remote terminals to explore the effects of VBR Groups of various alternative strategies before committing their group to a particular strategy.

Although the focus of this article is on educational usages, some of the facets of the urban electoral simulation model we are here describing would also be of obvious value to political planners. In fact, work has for some time been underway on a generalized urban electoral simulation model, called General Urban Election Simulation System (or GUESS), of which the computerized Woodbury simulation is a specific variant. In GUESS, urban electoral data specific to a particular city will constitute input to the general GUESS program.

In an article to be published elsewhere, I suggest that politicians, or public affairs decision-makers generally, have for different reasons much the same needs of rapid and conversational mode interactive feedback in simulation planning models as the educator.

At the heart of the computerized Woodbury simulation is the necessity of a time-share computer system which is reliable. Given the present state of the art, the reliability criteria implies having the program mounted on a back-up computer system. Unlike an all-computer simulation, mixed man-machine simulation systems such as the computerized Woodbury model involve a sizeable grouping of human participants who must have appropriate feedback at the requisite time in the simulation cycle, and significant delays in providing that feedback often distorts or destroys the model and/or the group dynamic which is a part of these exercises. In fact, it may well be that social scientists in the simulation field make more sophisticated demands on computer hardware than their "hard science" colleagues, because of their need for dependability of service. Additionally, the social scientist often works with sizeable collections of data, raising questions of computer core capacity and/or procedures to store data economically on tape and providing access to the tapes when required (at least one major computer time-share company we have worked with abhors hanging tapes, and the writer can testify from his experiences at several academic institutions that university computer centers are often reluctant to do so also.)

The challenges from the computer point of view, then, seem to be associated with questions of reliability (it's hard to tell some fifty simulation participants to "go home and come back later; the computer's died again," or "you'll have to wait for your budget reports until they fix the computer," and keep things rolling), input and output routines (for our purposes, time-sharing, conversational interactive modes, that is to say, computer queries for input, and computer responses in a English language format as easy for the user of the simulation model to understand as possible), and computer storage capabilities (either large capacity core and

disk/drum, as the UNIVAC 1108 or General Electric Mark II Time Share system, or peripheral and cheaper storage of the bulk of data on disk or tape, provided that the computer installation is willing to hang tapes to run a program with, let's say, at most five or ten minutes lead time, and that the computer system response as it appears to the user at the remote terminal, once the tapes are hung and the program under way, does not exceed, let's say, thirty seconds. In our experience with a cooperative computer center, these criteria imply that tape storage can be used in political simulation modeling. And/or, chaining of the program(s) may be undertaken, as is being explored with some of our simulation programs presently. Utilization of these techniques may well also open up this market to the mini-computer.

Urban Land Use Simulation Models

Several simulation models have been developed to explore alternative patterns of urban land use development. One of the earliest of these models, a fairly simple one useful for instructional purposes, is the aforementioned Community Land Use Game (CLUG) developed by Alan Feldt and associates at Cornell University.

The basic CLUG exercise is based on a grid-marked playing board, "playing pieces" to represent various types of urban construction (residential, industrial, commercial, etc.), and a player's manual describing the sequence of play in each round and other rules, and cost data for each type of construction, transportation costs, and so forth. Three teams, of several players each, represent both private entrepreneurial and collective community functions; representing the private sector of the economy, they seek to maximize their financial return by careful planning of their investments, and in their community role the teams must decide, by majority vote, upon the extension of so-called "utility segments" (representing functions such as police, fire, water, sewerage, and other such municipal services) to various parcels of land before private interests may build on them and realize profits. The utility segments, costing stated amounts to construct and to maintain each succeeding round, must be paid for out of community funds derived from the agreed-upon tax levies.

The model provides for variable land assessment patterns, to reflect changing land values brought about by development of surrounding parcels of land. A mechanism is provided to take into account depreciation in building values over time and a simple probability routine, which takes into account maintenance standards of various owners, is utilized to simulate the possibilities of loss of building use through fire, vandalism, or some other natural calamity. Transportation costs, varying according to type of material to be transported and class of roadway to be used, are calculated and charged to the appropriate team.

The simulation model is in actuality rather simple, although quite useful for instructional purposes, and it does produce "constructed" cities quite similar in appearance to many real-world cities. In any event, it does not come close to the complexity of the other available urban development models, such as METROPOLIS or REGION or CITY. Even so, however, to make the necessary calculations by hand for each round, and to record the necessary data for use in making following rounds' calculations, is a quite tedious job as anyone who has run CLUG for many successive rounds can testify. Additionally, we often find that the players are more pre-occupied with their complicated bookkeeping than with the facets of urban development we are concerned with teaching.

An additional problem is that it is not feasible, using hand calculation methods, to increase the complexity of CLUG much beyond its early status. Although we have added two more teams, and made a few other alterations in our work with CLUG, not much extension is possible because of the logistical problems.

Under these circumstances, it seemed reasonable to consider developing a computer program to perform the requisite calculations. And indeed, a computer program has already been developed at Cornell University; it is a batch-mode operation for the IBM 1130. Unfortunately for our purposes, however, it requires much attention for specific formatting on the keypunched input data cards, and also does not perform all the necessary calculations. Additionally, the resultant printout does not contain all the relevant data about the simulated community as thus far developed.

In short, two major flaws with the existent batch-mode program emerged: (1) again, as in the man-controlled mode of simulation play, the participants would be spending too much time and attention to questions of data format, at the expense of concentration on the dynamics of urban land use, and (2) the computer output was incomplete.

This being the case, it was determined to start anew, and to develop an appropriate computer program in a time-share, conversational mode environment since these facilities were available to us. The program as now developed "asks" questions for player decisions (buying land, construction of facilities thereon, utility extensions, etc.) in the correct order as appropriate to the simulation model. The questions are typed out in English on the teletype

unit, along with instructions for user response. The program calculates the appropriate cash flows and the new cash reserves for each team. A map is also printed out each round, detailing the land use configuration of the simulated community as thus far developed[9].

We find that the existence of the time-share computer program considerably assists in running sessions of a CLUG-type simulation (as we constructed the program, we made a number of modifications in the simulation model at the same time). Additionally, the computer vehicle now provides us with the option of adding various subroutines to the simulation model in our further work with this project.

From the point of view of computer technology, the criteria I raised earlier in this paper when discussing the urban electoral models pertain. Criteria of dependability, input and storage of large amounts of data, and conversational time-share modes predominate.

Computer Requirements For Political Simulation Modeling

Our approach to political simulation modeling to date has emphasized the use of large-scale time-share computing because that is what was available to us. There have been a number of problems in this approach, however. The problems usually center around the fact that many time-share computers do not at all live up to their billing in terms of response, time, and especially in terms of dependability. In part this may be an artifact of our doing most of our work at academic computer centers, and with one or two possible exceptions I have yet to see a university computer center which meets the criteria established in the preceding pages. And at least one of the major computer time-share companies in the commercial marketplace is customarily characterized by very slow (over several minutes) response time in the middle afternoons when demand on the system builds up.

Given the near unusability of many time-share computers for effective political simulation modeling, a recalculation of costs and benefits for the large-scale time-share computer and the mini-computer may be in order. For the typical cost comparisons seem to be based on the assumption that the larger machine does in fact what it is supposed to do. But given the realities of large machine performance, especially in the academic world, it may be that the cost comparisons do not in fact favor the larger machine.

The cost in programmer time, as well as computer time, lost in "bombed" runs on an ailing time-share system can be considerable (For one simulation class I know of, the professor ran up several thousand dollars worth of computer charges on his department at one university in attempting to mount his programs, finally gave it up as a complete loss, and went on during the next semester to refuse to teach the class at his home institution and to accept a part-time faculty position at a neighboring institution and teach the same course there—using that university's computer center.) Faculty members at more than one school have been known to leave for another university where computer service was better. And a recent Rand Corporation report on the efficiency and effectiveness of university computer centers suggests the efficiency situation with many academic computer centers is somewhat less than rosy.

Given the fact that the social sciences are increasingly coming to recognize the importance of computer analysis, and given the difficulty social scientists have in accomplishing their work with the large-scale computer at the university computer center, social scientists at a number of universities occasionally give thought to the possibility of acquiring a computer of their own, perhaps shared between several social science disciplines, and under their control. At least, so the thinking goes, perhaps some of the smaller projects could be put on "our own" computer, and the large jobs left up to the "number cruncher" at the university computer center.

The Mini-Computer To The Rescue?

Such thoughts have come to my mind a number of times when we have been having some of our more colorfully-exasperating sessions with the computer center, and so the opportunity to analyze these possibilities as a form of challenge to the industry in this article was welcomed. Given this opportunity, we have given some thought to the real nature of our requirements, and given that, to the ways our programs might perhaps be modified to accommodate the context of the mini-computer environment and also accomplish our goals.

First of all, we will assume that, given the small size of the mini-computer and consequently its small cost, it will be "controlled" by a relatively small number of users. That being the case, we will assume that we can dedicate its total use during a simulation run. This will accomplish, for our purposes, the same effect as time share.

The machine's typewriter unit, or perhaps an attached teletype, can be utilized to input data and commands, and to receive the results back, as the teletype unit on a time-share system does.

Obviously the major problem centers upon computer core storage capabilities. Here, some ingenuity is in order.

The present version of our TeleCLUG urban land use simulation game program takes some 32K of core on a UNIVAC 1108 for time share. While it is true that the present program is "sloppy" and "cleaning up" will shorten it considerably, we are still talking about something considerably larger than most mini-computer core sizes.

However, "chaining" of programs may enable an apparently larger program to be shoehorned into a smaller core. The Tele CLUG program, for example, was designed to be chained for a smaller computer, although that is not necessary for the present version on the UNIVAC 1108.

Also, the reader will recall that in earlier sections of this article I suggested that a response time of upwards of thirty seconds, or maybe even a little more, would not be a significant impediment to the work at hand. Considering that a good time share system talks about a response time of well under a second, this latitude should provide plenty of room to work in. In particular, with a dedicated machine, the use of disk storage, and perhaps some resort to tape (as, for example, the extensive data files in the Woodbury urban election simulation, as was done when that simulation was on the General Electric Time Share service) may prove to be a usable solution. In particular, if upwards of a thirty second response of the total machinery to the user is tolerated, considerable within-program disk referencing might be possible.

All of this may not seem "efficient" in terms of cost comparison with larger machines in the ideal sense, but when the fallacies of the larger machinery are taken into account, the dollar costs, as well as frustration aspects, may well rebound to the benefit of the mini-computer.

FOOTNOTES

1. Perhaps the leading example is the IBM Management Decision-Making Laboratory. See IBM Management Decision-Making Laboratory; Institute for Participants, and the companion, Administrator's Reference Manual, IBM, Technical Publications Department, 112 East Post Road, White Plains, New York, 1963.
2. Science Research Associates, 259 East Erie St., Chicago, Illinois. Researchers at Northwestern University are working with a computer-controlled variant of the INS, called the International Processes Simulation. Several faculty members at Rensselaer Polytechnic Institute and Northern Illinois University are working on the development of a multinational political, economic, decision-making simulation called PSW-1. This simulation, which is computer-controlled, provided for the generation of machine readable records of role-player decisions for subsequent analysis. (See John Parker, Clifford Smith and Marshall Whithed, "Political Simulation: An Introduction" in H. Ned Seelys, ed., Handbook for Teaching Latin American Cultural Themes, Illinois Department of Public Instruction, Springfield, Illinois, 1965, Ch. 5).
3. Colonel William Thane Minor, Director of the Simulation and Computer Directorate of the Industrial College of the Armed Forces in Washington has suggested in this regard that: "Economy of time, risk, manpower and dollars are key factors which make simulation useful. In the real world, it may be costly, or even impossible, to wait for feedback resulting from required decisions or for the results from suggested contingencies or selectable alternatives ... Decreasing risk, particularly where safety of people and property is concerned, has been a function of simulation for many years ... The economy of manpower is apparent in simulations where one individual may represent a group, an organization, or a political entity or where a small group of individuals may represent an entire nation ... (Finally) simulations which describe the 'state'; or changing 'states' of reality--usually cost fewer dollars than programs operating within reality for the same purpose." See Col. William Thane Minor, "Current and Future Uses of Time-Sharing in Educational Simulation," Simulation and Computer Directorate, Industrial College of the Armed Forces, Fort Lesley J. McNair, Washington, D. C., October 1968, pp. 4-5.
4. Allan G. Feldt, The Community Land Use Game, The Center for Housing and Environmental Studies Division of Urban Studies, Cornell University, Ithaca, N. Y., 1966.
5. The authors of the PSW Simulation are John R. Parker of IBM, Clifford N. Smith of Northern Illinois University, and Marshall N. Whithed. The model is discussed in Parker, Smith, and Whithed, "Political Simulation: An Introduction," in H. Ned Seelys, ed., Handbook for Teaching Latin American Cultural Themes, Illinois Department of Public Instruction, Springfield, Illinois, 1968, Ch. 5. The simulation is programmed in PL/1 for the IBM 360/40, batch-mode processing.

6. A manually-controlled (human umpire) version of this simulation model is to be published by Little, Brown and Company. The authors are H. Roberts Coward, Bradbury Seasholes, and Marshall Whitted. It deals with an urban mayoralty election campaign.
7. The Political Military Exercise was first developed by Goldhamer et. al. at the RAND Corporation; see H. Goldhamer and H. Speier, "Some Observations on Political Gaming, World Politics, Vol. XII, October 1959, pp. 71-83. See also H. Roberts Coward, The Political Military Exercise as a Teaching Device in Political Science: A Handbook, Final Report, Project No. 6-8964, Contract No. OEC-3-7-068964-2199, U. S. Department of Health, Education and Welfare, Office of Education, Bureau of Research, Washington, D. C., 1969.
8. Credit for making the developmental work possible must be extended to the General Electric Company, which provided the computer time and facilities for development of the prototype program. Mr. Robert Lund who, as senior programmer, devoted much "midnight oil" to the project, Mr. Lawrence Birch who served as research assistant, and Mrs. Cherie Campbell who assisted in developing the statistical format for the simulation model. Mr. Kirk Sorensen adapted the program to the UNIVAC 1108, which was used in the classroom exercise. Rensselaer Polytechnic Institute provided a Faculty Research Grant for carrying out some of the necessary work on the original development of the computer-assisted model.
9. The program, presently undergoing modification, was developed by Mr. Kirk Sorensen of Rensselaer Polytechnic Institute in the spring of 1970 and used in a class for the first time in May of that year at State University of New York College at Plattsburg. It is programmed in FORTRAN on the UNIVAC 1108, and is designed to be adaptable to other machines. Credit and thanks is due to the staff of the Computer Center of the State University of New York at Albany, which provided the computer facilities for this development. The land use map in this program is, as far as we know, the first one available over a teletype for this family of instructional urban land use models. Other models, such as CITY, provide a land use map, but only over a regular printer, and so the output must be carried from the computer site to the simulation room.

THEORY OF PROBABILITY AND STATISTICS
ILLUSTRATED BY THE COMPUTER

Elliot A. Tanis
Hope College
Holland, Michigan 49423
Telephone: (616) 392-5111

Background

During the past few years several Hope College seniors have done research projects in probability and statistics using our IBM 1130 computer. These projects often required computer simulation of either a physical experiment or a probability distribution. It became increasingly clear that in order for a student to write a computer program that would properly perform the simulation, the student had to understand the theory of the problem and the student also gained a greater appreciation for this theory.

It was then suggested that all students taking our two semester junior-senior level mathematical probability and statistics course should have this opportunity of increasing their understanding and appreciation of the theory by performing simulations on the computer. A drawback to this approach is the class time required and also the extra time required by a student outside of the classroom. It was also obvious that additional preparation time is required of the instructor of this type of approach.

It became clear that the addition of a two hour laboratory each week would provide the additional class time. In order to prepare materials for the laboratory and to provide additional preparation time for the instructor, it was decided to submit a proposal to the National Science Foundation. This proposal was funded. The grant from NSF is not only permitting us to develop materials for this laboratory, we are also developing materials for an introductory statistics course taught by Professor Herbert Dershen in which statistics and computer programming are interweaved in such a way that the student always has learned enough programming to program the next statistical technique and knows enough statistics so that the latest programming principle learned can be illustrated by examples involving previously learned statistics.

Description

During the summer of 1971 we developed software and wrote many of the laboratory exercises. We were assisted by four of our mathematics majors who had taken both the mathematical statistics course and the computer programming course.

During the 1971-72 academic year we are introducing the laboratory for the first time to a class of 24 students. An outline of the laboratories for the first semester is given in Section 3.

During the summer of 1972 we plan to develop additional software, revise our experiments and design some new experiments. We shall also be holding a one week conference in August for about 50 educators from the Midwest who are interested in these programs.

Following the 1972-73 academic year, during which the revised laboratory materials will be used with another class of students, we shall evaluate the entire project and make recommendations for the future. Hopefully we will be able to distribute materials during the summer of 1973 that have wide applicability.

Laboratory Description

The laboratory is used to show that certain physical experiments do satisfy particular probability properties. Some of these experiments are then simulated by the computer. An additional key purpose of the laboratory is for the student to understand and appreciate the theory of probability and statistics by writing computer programs. Thus we assume that our students have taken computer programming. We do provide a random number generator, various plotting subroutines, and additional subroutines when needed.

The following listing gives the topics covered in each of the 15 laboratories during the first semester. In Section 4 we shall describe two experiments.

Laboratory

Topics

1

Relation between a balanced spinner with a scale from 0 to 1 and a computer pseudo-random number generator.

- Probability as relative frequency - flipping a coin, a pair of coins or rolling a die - first done physically and then simulated using the pseudo-random number generator.
- 2 Permutations and combinations, hypergeometric probabilities. (Subroutines for drawing balls from urns and cards from a deck of playing cards were used.)
 - 3 Expectation.
 - 4 Conditional probability, Bayes' formula, independent events.
 - 5 Bernoulli, binomial, negative binomial, and geometric distributions.
 - 6 Empirical distribution for a single variable: histograms, ogives, empirical distribution function.
 - 7 Relation between probability density function and relative frequency histogram for the binomial and geometric distributions.
 - 8 Comparing the theoretical and empirical mean and variance for each of the binomial, hypergeometric, geometric, and negative binomial distributions.
 - 9 Poisson distribution.
 - 10 Empirical distribution for paired data.
 - 11 Multivariate, marginal, and conditional distributions for discrete random variables.
 - 12 Random number generators, distribution of $W = U + V$ where U and V are independent uniform random variables, distribution of $X = G(W)$ where $G(w)$ is the distribution function of W .
 - 13 Multivariate, marginal, and conditional distributions for continuous random variables.
 - 14 Generate random samples from the exponential, gamma, normal, and chi-square distributions.
 - 15 Generate random samples from the bivariate normal distribution.

Some of the topics which will be included in the laboratories second semester are the Central Limit Theorem, normal approximations, estimation, confidence intervals, power function, tests of hypotheses, analysis of variance, chi-square and Kolmogorov-Smirnov goodness of fit tests, and the sign test.

Laboratory Examples

We shall give a brief description of parts of two laboratories.

Laboratory 9: We shall describe an experiment for which the outcome has a Poisson distribution. We shall then show how the binomial distribution can be used to simulate a random sample from the Poisson distribution.

We often read in textbooks that if X denotes the number of alpha particles emitted by a radioactive substance that enters a prescribed region during a prescribed period of time, then X has a Poisson distribution. With assistance from the Hope College Physics Department; we took 100 .1 second observations of the number of emissions by BA 133 using a geiger counter in a fixed position. The students wrote a program to find the sample mean and the sample variance, showing that for the Poisson distribution these are about equal. These data are listed in Table 1. To visually compare the fit of the Poisson distribution to these data, a relative frequency histogram was plotted along with the probability density function for the Poisson distribution with a mean of $\lambda = 5.61$, the average of the 100 observations. This histogram is given in Figure

1. The subroutine that plots this histogram was written during the summer of 1971 by one of our students working on this project.

100 .1 SECOND OBSERVATIONS OF 8A 133.

7.	5.	7.	7.	4.	6.	5.	6.	6.	2.
4.	5.	4.	3.	8.	5.	6.	3.	4.	2.
3.	3.	4.	2.	9.	12.	6.	8.	5.	8.
6.	3.	11.	8.	3.	6.	3.	6.	4.	5.
4.	5.	9.	6.	10.	9.	5.	9.	5.	4.
4.	5.	6.	7.	7.	8.	6.	9.	4.	6.
5.	3.	8.	4.	7.	4.	7.	8.	5.	8.
3.	3.	4.	2.	9.	1.	6.	8.	5.	8.
5.	7.	5.	9.	3.	4.	8.	5.	3.	7.
3.	6.	4.	8.	10.	5.	6.	5.	4.	4.

THE SAMPLE MEAN AND SAMPLE VARIANCE ARE 5.6100 4.8978

TABLE 1

To simulate this experiment we used the fact that for large n and small p binomial probabilities can be approximated by Poisson probabilities. Earlier in the course students had learned to simulate random samples from binomial distributions. For this example we used the binomial distribution with $n = 100$ and $p = 0.0561$. Thus $np = 5.61$, the mean of the 100 observations in Table 1. A random sample of size 100 from this binomial distribution is given in Table 2. A relative frequency histogram of these data along with the probability density function for the Poisson distribution with a mean of $\lambda = 5.61$ is given in Figure 2.

A RANDOM SAMPLE FROM A $B(100, 0.0561)$ DISTRIBUTION.

8.	4.	8.	5.	6.	6.	7.	9.	9.	7.
4.	8.	5.	10.	5.	3.	4.	9.	6.	2.
8.	5.	2.	4.	10.	9.	7.	3.	7.	6.
5.	5.	7.	2.	5.	4.	8.	3.	7.	6.
6.	3.	5.	6.	5.	5.	4.	4.	5.	6.
6.	11.	6.	7.	7.	7.	8.	10.	3.	8.
6.	3.	6.	8.	2.	5.	7.	2.	7.	10.
2.	7.	5.	4.	6.	5.	7.	6.	4.	5.
12.	5.	7.	4.	2.	2.	7.	4.	6.	5.
6.	3.	4.	4.	4.	4.	4.	5.	11.	4.

THE SAMPLE MEAN AND SAMPLE VARIANCE ARE 5.7000 5.0099

TABLE 2

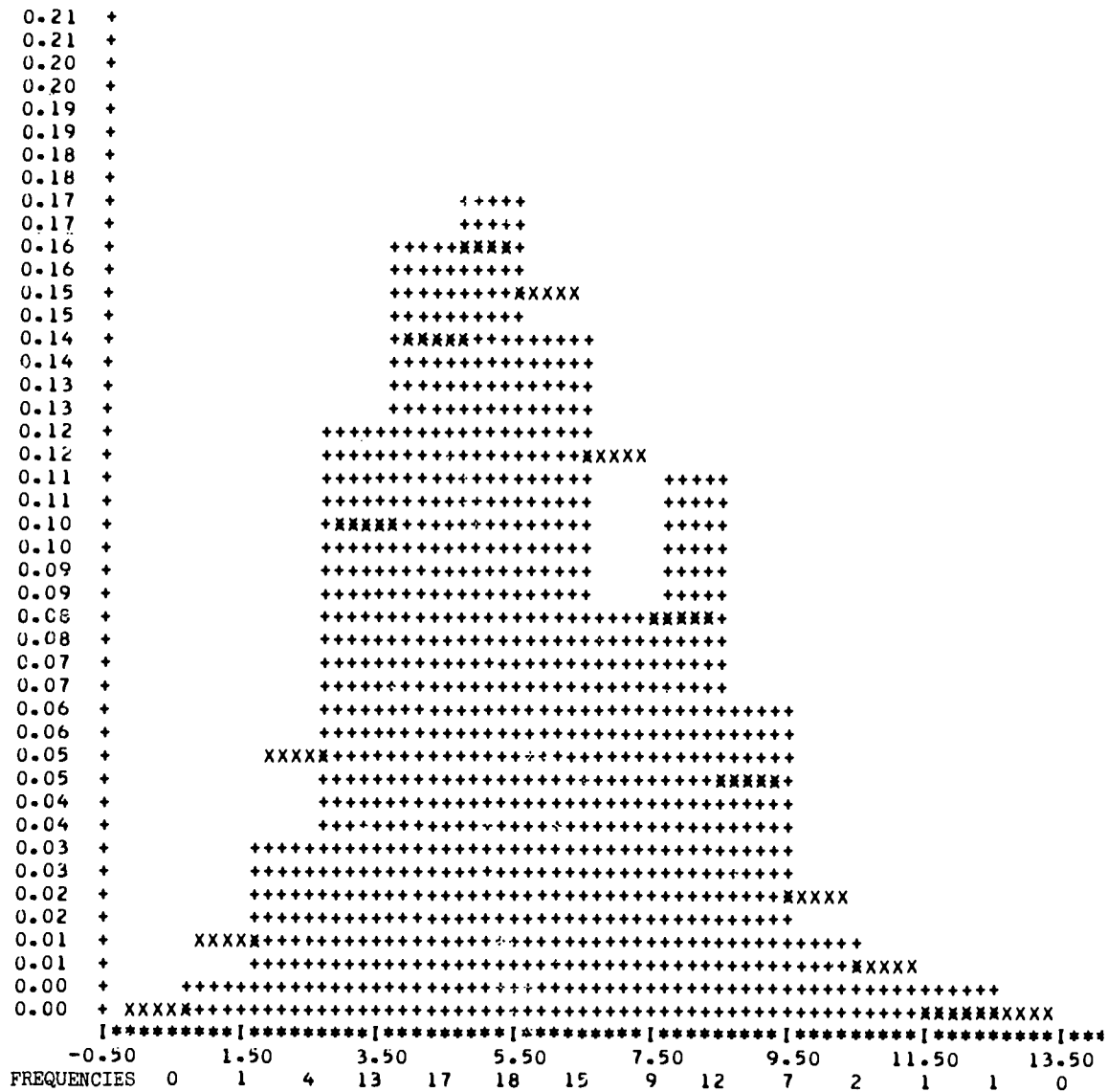


FIGURE 1

Laboratory 14: We shall describe a part of this laboratory that involves the exponential and gamma distributions.

The function

$$F(x) = \begin{cases} 0, & x < 0, \\ 1 - e^{-2x}, & 0 \leq x \end{cases}$$

is the distribution function of an exponentially distributed random variable with a mean of $\theta = 1/2$. The students are asked to simulate a random sample of size 500 from this distribution. They know that if Y has a uniform distribution on the interval $(0, 1)$, then $X = F^{-1}(Y)$ has an exponential distribution with a mean of $\theta = 1/2$.

Thus to empirically illustrate this theoretical fact, a sample of 500 random numbers is generated, say Y_1, Y_2, \dots, Y_{500} . For each $Y_i, X_i = F^{-1}(Y_i) = -(1/2)\ln(1 - Y_i)$. A relative frequency histogram for such a set, X_1, X_2, \dots, X_{500} , is plotted in Figure 3 along with the probability density function $f(x) = 2e^{-2x}$.

Now let V_1, V_2, \dots, V_5 be a random sample of size 5 from an exponential distribution with a mean of $\theta = 1/2$. The students know that $W = V_1 + V_2 + \dots + V_5$ has a gamma distribution with parameters $\theta = 1/2$ and $\alpha = 5$. That is, the probability density function of W is

$$g(w) = \frac{w^{5-1} e^{-2w}}{\Gamma(5) (1/2)^5}, \quad 0 < w,$$

zero elsewhere.

The students are asked to illustrate this theoretical fact. The sum of each consecutive set of 5 numbers in the sample of size 500 generated from the exponential distribution was found, yielding 100 w 's. A relative frequency histogram of these 100 w 's along with the probability density function $g(w)$ is plotted in Figure 4.

Evaluation

The reaction of students to the laboratory at this point is varied. One student said that he appreciated the lab because he realized that in order to write a program he had to really understand the theory. Students who had difficulty with programming were often frustrated by error statements which clouded their vision of the probability theory that was being illustrated.

In the first laboratories either too much material was covered or too many exercises were assigned. It became clear that it is better to illustrate less of the theory with three or four well conceived exercises. In this way the student has enough time so that many frustrations are avoided.

This laboratory would not have been possible without the previous summer for preparation, the released time for me during this academic year, and the student help in writing subroutines. We are grateful to NSF for providing this opportunity.



FIGURE 3

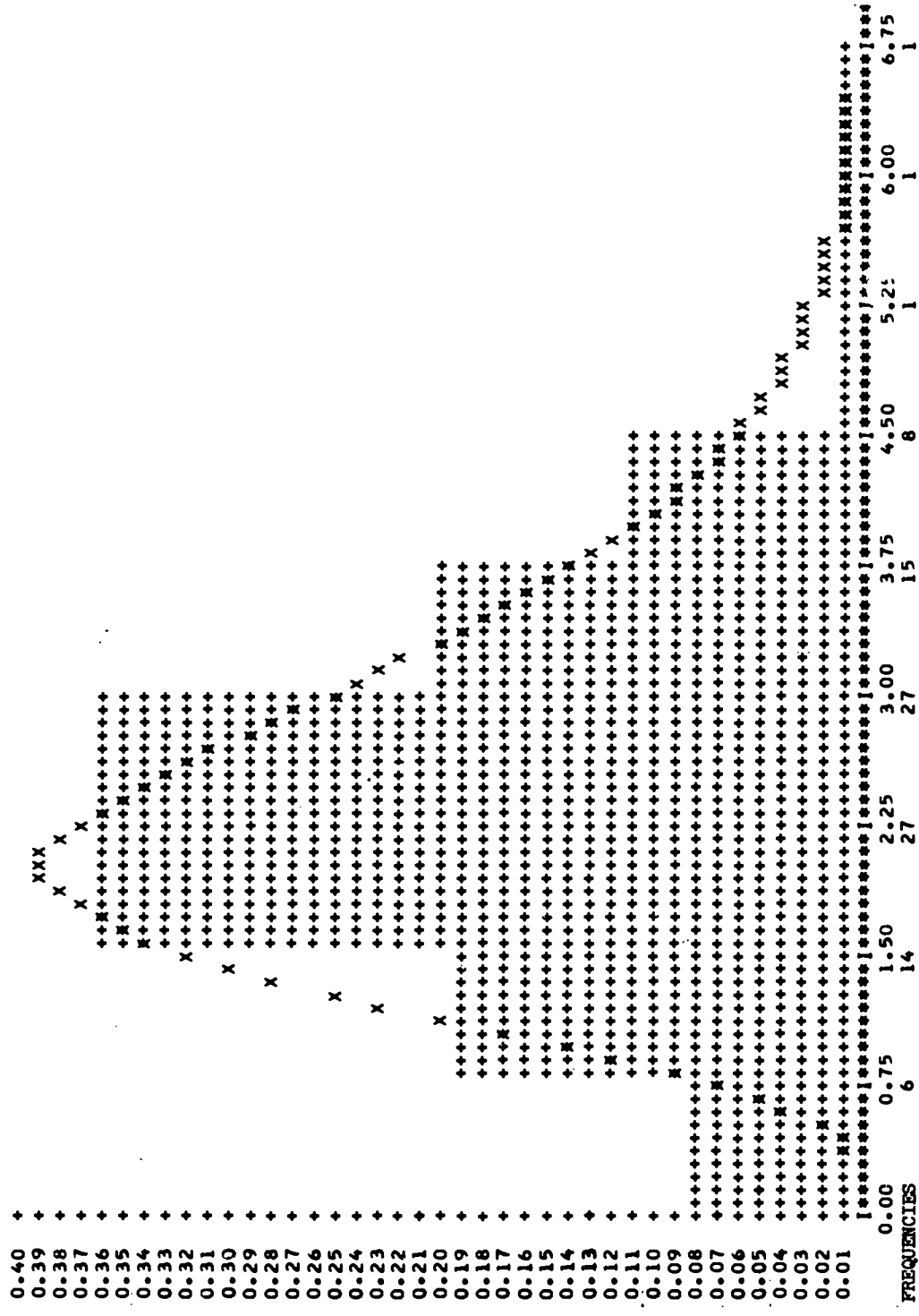


FIGURE 4

PSYSTAT - A TEACHING AID FOR INTRODUCTORY STATISTICS

Marc S. Weiss
Washington State University
Pullman, Washington 99163
Telephone: (509) 335-4915

Papers presented at the previous two conferences (e.g. Koh, 1970; Wikoff, 1970; Sandusky, et al, 1971) have described the problems associated with traditional approaches to teaching introductory statistics to behavioral and social science undergraduate majors. The use of a computer statistics laboratory is an attempt to resolve these difficulties which may be briefly restated as follows:

1. Students have weak (if any) real skills for handling quantitatively oriented material and often have feelings of "dread" when they enter such a course.
2. Traditional emphasis on hand and desk calculator computations and computational formulae often results in neglect of theoretical and conceptual material.
3. Absence of any meaningful laboratory experiments or demonstrations handicaps the development of an "intuitive" feel for basic statistical theory.

In addition to these problems, I was faced with two additional ones when confronted with the prospect of teaching the psychology department's introductory statistics course. (1) The lack of an appropriate text for a computer-oriented course and (2) the lack of appropriate software easily implementable at the WSU computing center. Consequently, I undertook the development of a software package (PSYSTAT) with the following goals in mind:

1. The package should be sufficiently general purpose so that it could be used with any suitable standard text book.
2. The package should be easily transportable (written in FORTRAN) and modular in design to allow for easy modifications and expansion.
3. It should be easy to use, require minimal learning effort on the part of the student (e.g. formatting of input should be consistent regardless of options selected) and should require no programming effort on the part of the student.
4. The programs should incorporate extensive diagnostics to catch student errors.

Since the WSU psychology department has no remote facility, the PSYSTAT package was designed to be used in a batch processing environment.

Description of Course

The introductory statistics course in the WSU psychology department is a sophomore-junior level course taken mostly by psychology majors. However, approximately 30% of the students are non-majors. There are no prerequisites other than high school algebra. The course content is traditionally devoted to both descriptive and inferential statistics. The topics covered when PSYSTAT was used included frequency distributions, descriptive statistics using grouped and ungrouped data, basic probability, the normal distribution, sampling distributions, central limit theorem, parameter estimating and confidence intervals, t-test, correlation, introductory experimental design and some distribution-free statistics.

What follows is a description of the course given to students on the first day of class and outlines the basic course philosophy and organization:

ELEMENTARY STATISTICS - INTRODUCTION

"The Purpose of Computing is Insight not Numbers" - R. W. Hamming, 1962

This could well be rephrased "The purpose of statistics is insight not numbers." This course is designed with this goal in mind. Statistical insight comes with practice and experience and inevitably involves computation. Unfortunately, the beginning student often gets lost in a maze of arithmetical computation and manipulation. Formulas and desk calculators provide little time for the development of insightful understanding.

Modern technology has provided a tool in the form of the high speed digital computer which not only can eliminate the need for hours of hand calculation but also opens up a whole new field of computer simulation. It is through this latter use of the computer as a tool for experimentation that the hoped for insights will be developed. No programming skills are needed

and none will be taught. The student will complete two types of assignments: (1) hand worked ("paper and pencil") problems, and (2) computer worked ("machine") problems.

The paper and pencil problems are designed to familiarize the student with simple applications of basic statistical concepts and theory. The machine problems are designed to enable the student to perform "experiments" and compare results with the statistical theory. In order to complete the paper and pencil problems, a student must:

1. Know how to write on paper with a pencil (or pen).
2. Know simple arithmetic.
3. Learn some statistics.
4. Be able to think.

In order to complete the machine problems, a student must:

1. Learn how to write on "IBM" cards with a keypunch.
2. Read some simple instructions.
3. Learn some statistics.
4. Be able to think.

The point of all this is that "using the computer" should frighten no one. All students will receive instruction in keypunch operation. A handout supplement will also describe use of the keypunch.

Classroom dittoed supplements will be made available periodically, describing those aspects of the course not covered by the text. There are no exams in the course. Each Tuesday students will receive an assignment to be completed and returned the following Tuesday. No late assignments will be accepted. Each assignment will have a maximum point value. A student's final grade will be determined by the total points he or she has accumulated. The grading system will be discussed in class, but in general will be based on the individual score, not the distribution of scores (if you don't understand this last sentence, read it again in two weeks).

Description of PSYSTAT

The PSYSTAT package was developed not only to meet the challenge of improving statistics instruction but to make a student's first contact with the computer a minimally traumatic experience. To meet these goals the following features are incorporated in the package:

1. Input Data - these may be student supplied (all data unformatted with one datum per card) or generated by the student via simulated sampling from one of four (or more) populations. The populations available initially were the normal, binomial, uniform, and one called WEIRD which was asymmetric and bimodal.
2. Control Cards - a minimum of PSYSTAT control card information is required. The following three cards are always needed: an Identification Card which labels the output with the student's name and any other information he wishes to supply; a Program Name Card which controls which of the 4 basic programs (DESCRIBE, T-TEST, CORRELATION, STATISTICS) the student wishes to use; a Data Card which identifies the source of data as being cards or a machine simulation.
3. Basic Programs - (These four basic programs are being currently supplemented with programs to do simple analysis of variance, chi-square and non-parametric statistics).

DESCRIBE: The basic descriptive statistics program. Input is either grouped or ungrouped data. Output contains a frequency histogram, a table of intervals, frequencies and cumulative frequencies as well as a table of summary statistics (mean, median, standard deviation, range, quartiles, etc.)

STATISTICS: Can be used with any other program (or alone) and generates summary statistics for input data.

T-TEST: Performs one of three t-tests depending on input information. The output contains sample size, degrees of freedom and the resulting t statistic.

CORRELATION: Performs simple linear regression. Output contains the sample size, correlation coefficient, two regression equations with standard errors for each of the coefficients.

4. Error Diagnostics - Extensive error detection and error information capability has been built into the PSYSTAT package. This enables the student (and instructor) to diagnose the cause of an abnormal program termination. As a result of my experience this past semester, this feature is being expanded even further.

Sample Problems

Below are two problems, one from the first (descriptive) part of the course and the other from the later (inferential) part of the course. It should be noted that the basic simulation for the first problem requires a total of 5 cards or lines of input and 6 cards or lines for the second. Other sample problems and typical student results will be presented during the conference.

Sample Problem #1

Purpose: To test the mode, median and mean for stability.

Question 1: Which of these measures of central tendency vary most (and least) from sample to sample?

Question 2: Which of these measures is most (and least) stable when they are computed from grouped data?

Method: Choose two of the four random distributions available in PSYSTAT. Each distribution uses N's (sample sizes) of 20, 100, and 500 and the following number of intervals:

<u>N</u>	<u># int</u>		
20	3	6	8
100	5	10	15
500	10	20	30

That is, for each of the two distributions, a minimum of nine machine computations must be made.

Results: Using the form provided (for each distribution) enter the appropriate sample median, sample mean, grouped mode, grouped median and grouped mean. Hint: You may want to use the computer to calculate the grouped median and mean.

Discussion: On the form provided, discuss your results. What can you conclude about the relative stability of these three measures of central tendency? Are the results the same for both distributions?

Sample Problem #2

Purpose: To test the effect of significance level and sample size on the accuracy of the one sample "t" test. The probabilities of Type I and Type II errors will be empirically investigated.

Method: Type I error is defined as the rejection of a true hypothesis. Type II error is the failure to reject a fake hypothesis.

- Use the NORMAL population (mean = 0) to generate samples of different sizes. Use T-TEST to test a true hypothesis (H_0) about the population mean. Run a sufficient number of simulations for each sample size so that you can plot the probability of Type I error as a function of sample size (N) for each of three significance levels.
- Repeat 1, using a false hypothesis (H_0). Plot the probability of a Type II error as a function of N for each of three significance levels. Also plot $1 - p$ (Type II error) as a function of N for each of your three significance levels. These last curves are the "power" curves for the "t" test for your choice of hypothesis.

Results: Discuss your empirical curves. What effects should significance level have on probability of Type I and Type II error? Do your results confirm your expectation? If not, why? What effects should sample size have on these probabilities? Are your results in agreement? If not, why? Are the assumptions underlying the "t" test violated in this experiment? If so, which assumptions?

Results

Since there has been only one semester's experience with PSYSTAT, quantitative assessment of the effect of its use was not undertaken. Students were asked to evaluate their experience and the results of this evaluation are entirely consistent with my own judgment. Approximately 90% of the 87 students felt that using PSYSTAT was worthwhile for one or both the following two reasons:

1. It eliminated concern with hand calculations and computational formulae.
2. It eliminated the air of mystery surrounding the computer's capabilities and functions.

In addition, a majority (60%) felt that their comprehension of the course content was enhanced by using PSYSTAT and that the simulation experiments were worthwhile. The remaining 40% either found no value at all in the computer exercises or were equivocal in their response.

On the negative side, complaints were mostly due to the nature of batch mode operation: excessive time spent in wasted trips to the computing center (long turn-around) and long waits at the keypunches. There were also the unavoidable complaints about down time. The use of remote terminals will hopefully reduce some of these problems and these are being included in future plans.

The current cost of using PSYSTAT in the batch mode is quite low. The 87 students in the class each used an average of 5 minutes of machine time (IBM 360/65) during the semester with an average cost of \$20 per student per semester. This figure will hopefully be even lower when planned improvements in PSYSTAT's efficiency become implemented.

REFERENCES

1. Koh, Y. O., "The computer is an instructional tool for the statistics course," Proc. Conference on Computers in the Undergraduate Curricula, 1970, pp. 2.4-2.17.
2. Wikoff, R. L., "Using the computer in basic statistics courses," Proc. Conference on Computers in the Undergraduate Curricula, 1970, pp. 2.18-2.24.
3. Sandusky, A., Campos, F., Livingston, J., Lovejoy, E.P., Messick, D. M., "Instruction in Statistics: A Report on the Computer Laboratory for Analysis of Data in Psychology." Proc. Conference on Computers in the Undergraduate Curricula, 1971, pp. 429-436.

A COURSE ON COMPUTING AND STATISTICS FOR SOCIAL SCIENCE STUDENTS

Herbert L. Dershen
Hope College
Holland, Michigan 49423
Telephone: (616) 392-5111

Origin of the Course

Before the introduction of the course to be described in this paper, most economics and social science students at Hope College enrolled in two mathematics courses. These were introductory statistics and computer programming. In the academic year 1969-70 two developments occurred which pointed to the advisability of combining these courses.

First, Dr. Jay Folkert conducted experimental sections of the introductory statistics course at Hope College in which he used the computer as a tool to obtain illustrative information. He did this by allowing students to use previously written programs on prepared data decks. This was done in conjunction with the project to study the use of the computer in statistical instruction sponsored by the National Science Foundation and the University of North Carolina, in which Dr. Folkert was a participant. At the close of the experiment it was Dr. Folkert's feeling that the computer was an asset in such a course but that something was lost because the students were not able to participate in the preparation of the programs.

At the same time, a course on computing for social science students was introduced into the Hope curriculum. This course is basically a FORTRAN programming course. Those who were involved with teaching this course found that some knowledge of statistics would be valuable to those students enrolled. With some statistics background, the students could be assigned projects pertinent to their fields of interest.

It was therefore the consensus of the Hope College mathematics department that it would be more valuable to combine statistics and computer programming into one course for social science students rather than to continue with two separate courses. A proposal was made in the fall of 1970 to the National Science Foundation for the development of such a course, in conjunction with the development of a laboratory for the mathematical probability and statistics course, and this project was funded.

Value of the Computer

There are three major reasons that the computer is an asset to the introductory statistics course. First, exposure to the computer and computing is a necessary experience for any social science student. He should be aware of the economic, sociological and psychological impacts of the computer as well as the application of the computer to the solution of problems in his discipline.

Second, the computer can serve as an aid in teaching statistics theory. The student has a much greater mastery of a concept after he has explained it to someone else. When a student writes a program he is doing just this, explaining to the computer how to solve the problem. For example, in the assignment shown in Figure 3, the student is asked to explain to the computer how to test hypotheses. Also the computer can be used to provide the student with illustrations which further enhance his understanding. An example of this is found in the assignment given in Figure 2 in which the student is asked to illustrate the normal approximation to the binomial.

Third, the computer allows the student to apply the statistical procedures he is learning to useful sets of data, thus giving him valuable experience in interpreting results and an interesting incentive for learning the statistics.

Description of the Course

The course being developed is entitled "Applied Statistics and Computer Programming." The only prerequisite is high school algebra. It is a two semester course for three hours credit each semester. An outline of the topics presented in the course is found in Figure 1.

The students are introduced to a simplified form of input and output so that they can begin programming without being exposed to FORTRAN statements. This is done by subroutines written for this course because Hope College has an IBM 1130 which has no simplified input/output included in the system. The author has prepared notes to serve as a text for the class for the FORTRAN portion of the course because existing texts present the language in a sequence different from that determined to be optimal for this course. For example, we present subscripted variables very early in the course because they are needed to program examples and procedures in descriptive statistics.

First Semester

1. Introduction to computers - Computers, algorithms, lang.
2. Elements of FORTRAN - Constants, variables, assignment statements, transfer statements, simplified input/output.
3. Probability - Rules of probability, counting techniques
4. Subscripted variables
5. Descriptive statistics - Frequency distributions, mean, standard deviation, other measures.
6. Probability distributions - Discrete, continuous, Chebyshev's inequality, Binomial, Normal.
7. Random sampling - Random number generators, sample means.
8. Subprograms - Functions, subroutines.
9. Other computer languages and systems - COBOL, list processing languages and their applications, time-sharing, man-machine interaction.

Second Semester

1. Input/Output - FORMAT statements.
2. Estimation - Confidence intervals, t-distribution.
3. Testing Hypotheses - Type I and II errors, testing means, testing proportions.
4. Correlation and regression - Multiple linear regression, non-linear regression, interpretation of results.
5. Multiple subscripts
6. Chi-square - Contingency tables, goodness of fit.
7. Analysis of variance - use of library programs.
8. Nonparametric tests - Sign test, rank-sum test.
9. Miscellaneous - Implications of computers, experimental design, simulation.

Figure 1. Outline of the course.

Purpose: The purpose of this assignment is to illustrate how the normal distribution can be used as an approximation to the binomial, when the approximation is good, and how to use the normal table program XNORM.

Description: Write a program which reads n and p and using subprogram BINOM computes the binomial probability that $x = k$ for $k = 0, 1, \dots, n$, and the normal probability that $k - \frac{1}{2} \leq x \leq k + \frac{1}{2}$ for the normal distribution with mean np and variance $np(1-p)$, and the same values of k . Use XNORM described in Appendix E to find normal probability.

Output: The output is to consist of one line containing n , one containing p , followed by $n+1$ lines each containing a value of k , the corresponding binomial probability and the normal probability.

Questions: 1. Try out your program for a variety of values of n and p . Do the cases where np and $n(1-p)$ are both greater than 5 show good accuracy? How is the accuracy when the above rule is violated?

2. Does the accuracy of the approximation tend to vary with k for fixed n and p ? If so, how?

3. Do you notice that one probability is always larger than the other? Can you explain this?

Extra things to Try: Write a program which is the same as the one described above but which computes the probability that $x \leq k$ instead of $x = k$. Answer questions 1-3 for this program.

Figure 2. Sample Assignment
Normal Approximation to the Binomial

Purpose: The purpose of this assignment is to introduce the student to testing a hypothesis about a sample mean and illustrate type I and type II errors.

Description Write a function subprogram which has the following arguments: XMU0, the hypothesized mean, XMU, the actual mean, SIG the actual standard deviation, and N, the sample size. The subprogram is to generate 100 samples of size N from a normal distribution with mean XMU and standard deviation SIG. For each sample, a 95% confidence interval is constructed assuming sigma known, and a test is made as to whether XMU0 is in the confidence interval, i.e., whether $\mu = XMU0$ is accepted. A count is made of the number of times the hypothesis is accepted. This is the value to be returned for the function.

Write a calling program which calls this function four times, each time with XMU0=20, SIG=5, N=10, and for XMU=20,22,25,30.

Output: Your output should consist of XMU0, XMU, SIG, N and the value of the function for each call of the function.

Questions: 1. Relate the results of each call to function to either type I or type II errors. Specify which.

2. Punch a card summarizing your results.

3. What would be the effect on your answers if SIG were 10 instead of 5? What if N were 20 instead of 10? What if we used a 99% instead of 95% confidence interval?

4. Compute the theoretical probability of making an error in each of the four performed tests of hypotheses. Indicate for each whether it is a type I or a type II error. Compare these with your results.

5. The above program tests the hypothesis $\mu = 20$ against the two-sided alternative $\mu \neq 20$. How would your answers be changed if a one-sided alternative $\mu > 20$ were used? What if $\mu < 20$ were the alternative?

Extra things to try: Add enough generality to your program that you can try some of the things suggested in questions 3 and 5 above.

Figure 3. Sample Assignment Testing of Hypotheses

520

The statistics text chosen is Elementary Statistics by Paul G. Hoel, and the topics covered follow the presentation of the text with exceptions noted below. Descriptive statistics and probability are reversed in order that the students may gain some familiarity with PORTMAN and subscripted variables before they are needed for descriptive statistics. Some discussion is added concerning random number generators along with experience in their use when random sampling is treated. Also, the students are given practice in using canned subroutines and interpreting their results for regression and analysis of variance.

Five data sets, each consisting of several variables, are stored on a disk. The students learn early in the course how to access this data. These sets are data actually used for research purposes in the areas of sociology, psychology, education and economics, and have been contributed by faculty on the Hope campus from their research and from other books and articles. Already three additional sets have been contributed for next year. Each student is assigned one data set and one variable from that set which he uses throughout the year. This use ranges from finding the mean and standard deviation to taking random samples to obtain confidence intervals to correlation and regression with other variables in the same data set.

The students work fewer textbook problems than in the standard statistics course. Instead, they write computer programs for solving these problems and then apply these programs to their data sets. The students are given a total of 25 assignments involving the computer throughout the year. This averages out to slightly less than one assignment per week. The assignments typically involve the writing of a program, the application of that program, and the answering of some questions intended to bring out the important points. In many cases students are asked to punch cards summarizing their results so that a mean result may be obtained for the entire class. Extra credit problems are given along with each assignment to challenge the better students. Two sample assignments are found in Figures 2 and 3.

Description of the Project

The project for developing this course is of two years duration. In the summer of 1971 this course was organized with the assistance of four senior mathematics majors who wrote the necessary subroutines and programming examples as well as testing the computer assignments. The course is being taught for the first time in the academic year 1971-72 with a starting enrollment of 26 students. The summer of 1972 will be devoted to revising the course according to the experience gained during the preceding academic year. The course will then be taught in revised form during the academic year 1972-73 and the following summer a final report will be made along with the final preparation of materials such as course outline, lecture notes and assignments. These materials will be distributed to allow other staff members to teach the course.

Results

At this writing it is the middle of the first year of the project and hence too early for any firm discussion of results. Thus far the reaction of the students has been most favorable. Some have indicated that they feel the statistics is made easier to understand by the use of the computer. I suspect, however, that there are others for whom the computer simply clouds the issue. More students have been completing the extra credit portion of the assignments than was expected.

The author feels that the morale and interest of the students are much greater in this course than in either of the two parent courses, and for this reason, the course is a pleasure to teach. There has also been a favorable response to this course from our social science faculty who feel it is a most valuable course and are encouraging their students to take it as well as assisting us in its development.

AN ALTERNATIVE APPROACH
IN
TEACHING STATISTICAL METHODS

S. C. Wu
California State Polytechnic College
San Luis Obispo, California 93401

An undergraduate course in statistical methods is generally a non-mathematical exposition of the theory of statistics. The statistical techniques covered in such a course are mostly, if not completely, based upon the normal theory. Because the normal distribution is a continuous distribution, an in-depth study of the distribution necessarily requires the background of calculus. Hence, in a pre-calculus methods course, even the simplest techniques based upon the normal theory cannot be rigorously justified. Consequently, it is difficult for the students to grasp some of the important statistical concepts and to understand the limitations in applying the normal theory. These difficulties cannot be completely eliminated without relying on the concept of the randomization theory.

The randomization concept and the theory of significance tests based on it appear to be entirely the results of R. A. Fisher's search for the underlying principle of experimental design (1926). The ideas involved are very simple. The null hypothesis, used in all treatment-comparison experiments, is that there are no differences of any kind in the effects of the various treatments on the responses of those experimental units under study. If this is held to be true, and if the treatments are assigned randomly to the experimental unit, then all possible associations of treatment labels and experimental-unit responses are equally probable. For every association, one can calculate (at least conceptually) the chosen test statistic and derive its distribution from the equally probable associations. Then it is a simple matter to calculate from this distribution the probability of any outcome as favorable or less favorable to the null hypothesis as the one observed. This is the significance probability.

A simple worked example of the randomization test can be made using the data from G. W. Snedecor's (1956) Statistical Methods. Two preparations of the Mosaic virus were tested on leaf number 8 of eight different plants. Each test leaf represented a pair of sampling units, separated by the leaf mid rib. Treatments were assigned at random within responses. The data were:

		Lesions per Half Leaf								
Plant number		1	2	3	4	5	6	7	8	Mean
Preparation	A	9	17	31	18	7	8	20	10	15
Preparation	B	10	11	18	14	6	7	17	5	11
A-B		-1	6	13	4	1	1	3	5	4

If the null hypothesis of no differential effects of the virus preparations on the production of lesions is held true, then this is but one of $2^8 = 256$ equally probable, possible sets of differences. Those equally favorable to the null hypothesis and showing an excess of lesions for preparation A are:

1	6	13	14	-1	1	3	5
1	6	13	14	1	-1	3	5

and that one less favorable to the null hypothesis is

1	6	13	14	1	1	3	5.
---	---	----	----	---	---	---	----

These four and their symmetric complements obtained by switching labels A and B on each and every leaf are the eight possible outcomes which deviate as much or more from the null hypothesis as does the outcome actually observed. Thus, the significance probability is $8/256 = .031$. Under the null hypothesis, the probability of observing a difference as much or more than the difference actually obtained is small (.05 is the usual critical significance probability), so we place in doubt the hypothesis of no differences between the two preparations of the Mosaic virus. Thus, the test is justified step by step. A simple paired-comparisons t-test based on the normal theory gives computed (t_c) = 2.63 and significance probability of

$$P(|t_{(7)}| \geq 2.63) = .034.$$

The agreement between the two ways of calculating significance is remarkable. In fact, since t_c is a monotone function of \bar{x} , $\bar{x} < \bar{x}'$ if and only if $t_c < t'_c$, we may write

$$P_R(|t_{(7)}| \geq 2.63) = .031$$

$$P_N(|t_{(7)}| \geq 2.63) = .034$$

where R and N stand for: "derived from the randomization distribution induced by randomization of treatments to experimental units" and "derived from sampling independent identically distributed normal responses." Since the derivation of the t -distribution and the computation of significance probability are far beyond the scope of an undergraduate course in statistical methods, the t -test cannot be rigorously justified. Another good feature of the randomization theory is that the role of randomization in experimental design becomes clear. On the other hand, should all conditions for a valid test based on the normal theory be satisfied, the randomization procedure in an experiment can be omitted. Hence, within the framework of the normal theory, it is extremely difficult to explain the role of randomization. According to the randomization theory, it is also easy to show why a small sample cannot provide enough information for a valid test. For instance, if a sample of size two is used to test a hypothesis about a mean, the smallest possible significance probability is $2^{-2} = .25$ which is much too large to be regarded as significant. Similarly, the smallest possible significance probability for testing hypothesis about equal means in a 3×3 Latin square design is $.5$. However, within the framework of the normal theory, it is difficult to show that these tests may be totally invalid.

Currently, the randomization test is seldom taught in a statistical methods course because it is not easy to evaluate P_R in general. Each time a randomization test is used, the critical region must be determined on the basis of the data which were collected. To describe the degree of difficulty involved, consider an analysis of variance problem with 8 treatments, 2 blocks and 8 plots per block. For these, the possible ways the treatments can be assigned to plots number $(8!)^2$. Since $(8!)^2$ is in excess of 1.25×10^9 , it is not feasible to enumerate all possible assignments of treatments. Without a high-speed computer, the application of the randomization test is limited to extremely simple problems. Using a high-speed computer, one can sample (Monte Carlo) the population of possible randomization 1000 times with replacement and calculate sample estimates of significance probabilities. Errors attributable to this sampling have a binomial standard deviation

$$\sqrt{\frac{p(1-p)}{1000}} = \frac{\sqrt{p(1-p)}}{31.62}$$

In the range of p , usually regarded as significant ($p \leq .05$), this is at most $.006891$ and, hence, of slight practical concern. The maximum of the standard deviation is $.01581$ which occurs at $p = .5$, a value of no practical interest in testing hypotheses. Thus, the application of the randomization theory is broadly extended. The standard deviation of the estimator is always a constant divided by the square root of the sample size. As the sample size increases, the error attributable to this sampling decreases. A worked example of the Monte Carlo method applied to a three-factor factorial design with replicates (subsamplings) was given by Wu and Williams (1971).

The idea behind the simulation is essentially the estimation of a proportion of a binomial population. Since estimating a proportion represents a class of important problems which are usually covered even in a statistical methods course at lower division level, simulation introduces no new concepts. High-speed computing has gradually become an integral part in teaching statistics. The use of a computer in simulation will not create any additional difficulties. Perhaps it is to the advantage of the students to expose themselves to high-speed computing at the earliest possible time.

It has been shown, generally, by Urquhart (1965) that the significance probabilities derived from randomization tests are well approximated by the significance probabilities derived from normal theory. Hence, the usual statistical methods can be introduced as approximations to randomization tests. The new approach has the theoretical elegance of the randomization theory and retains computational simplicity of the tests based on the normal theory. At present, the randomization theory still cannot replace the normal theory in an undergraduate course in statistical methods because of high cost of computer time, but it can certainly supplement the normal theory. It is well known that, if any of the conditions for a valid test based on the normal theory is not satisfied, the consequence may range from low power to grossly invalid test. When in doubt, one should use the randomization test instead of a test based on the normal

theory. The randomization test is a nonparametric test but, unlike other nonparametric tests, it is a powerful test. According to Lehman and Stein (1949) and Hoeffding (1952), the asymptotic relative efficiency (A.R.E.) of a randomization test is 1.0 when compared to the most powerful parametric tests in some specific situations.

In conclusion, it is shown that the randomization theory can be easily incorporated into a statistical methods course. Many advantages of the simple theory make the short time required to cover it worthwhile. Using this new approach in teaching, the pioneers can expect some resistance from the students because no adequate textbooks or reference books are available, and the randomization theory is in addition to the standard methods that must be studied. However, the resistance will diminish as the new approach becomes widely accepted and the students see the advantages of the new approach.

REFERENCES

1. Fisher, R. A. (1926). "The arrangement of field experiments." Jour. Ministry Agric., 33, 503-513.
2. Hoeffding, W. (1952). "The large-sample power of tests based on permutations of observations." The Annals of Mathematical Statistics, 36, 369-400.
3. Lehmann, E. L. And Stein, C. (1949). "On the theory of some nonparametric hypotheses." The Annals of Mathematical Statistics, 20, 28-45.
4. Snedecor, G. W. (1956). Statistical Methods. Iowa State College Press, Ames, Iowa, 49-51.
5. Urquhart, W. S. (1965). "On the permutation distribution of a multivariate test statistic." Ph.D. dissertation, Colorado State University, 106 pages.
6. Wu, S. C. and Williams, J. S. (1971). "Randomization test for factorial designs." Proceedings of the Fifth Annual Symposium on the Interface of Computer Science and Statistics. Oklahoma State University, Stillwater, Oklahoma.

INDIVIDUALIZED INSTRUCTION IN BASIC STATISTICS:
AN EXPERIMENT IN COMPUTER MANAGED INSTRUCTION

Frederick S. Halley
State University of New York
Brockport, New York

Introduction

The purpose of this paper is to describe how computer batch processing was utilized in the development of a programmed self instructional course, its costs, student evaluation of the instructional method, and future plans for the continuation and development of computer managed self instruction.

During the past three years, a team of instructors at S.U.N.Y. Brockport have been developing and applying a programmed self instructional technique to the teaching of basic statistics which presents a significant departure from the usual lecture/discussion method of instruction. The technique surrounds students with all of the necessary resources to master course objectives and turns them loose to do it on their own without traditional classes. In this manner, students are required to become active participants in the learning process and may learn by doing the activities they are usually only told about in traditional classroom situations. The improvement and individualization of instruction is made possible by the use of computer technology in the management of instruction.

Description of the Course

The structure of the course presents a maximum amount of freedom for the student. The student may select from several alternatives to meet the requirements of the course. His only requirements are to turn in homework assignments and pass a final examination. Materials available to the student for use in meeting course requirements include a self-study guide, a programmed statistics text, homework assignment sheets, an individualized set of data, audio notebooks and accompanying visual aids, and a statistics laboratory equipped with calculators. Homework assignments, due periodically during the semester, set a minimum speed at which students' work progresses. Nothing prevents a student from expediting this work to complete the course ahead of schedule. Students have taken the final examination as early as midsemester.

The student study guide provides learning parameters for the student as well as instructions for learning. In the study guide (Nasca, Potter and Halley, 1971) there are twelve units, each with a homework assignment. Each unit states behavioral objectives which the students should be able to accomplish after completion of the unit, a summary of principles and concepts used in the unit, examples, and references to other available instructional materials, such as programmed texts, audio notebooks, and transparencies. The units cover levels of measurement, central tendency, variability, score transformations, the normal curve, sampling distributions, the t-ratio, correlation, regression, prediction and chi square. The study guide is not intended as a substitute for a text, but rather a device to provide direction to the student in his endeavor to learn statistics.

For each unit, the student is provided with a homework sheet which contains exercises specific to the unit. It requires the student to demonstrate accomplishment of the behavioral objectives of the unit by applying the principles and concepts of the unit to his own individualized set of data. While each student has the same assignment, different sets of data prevent plagiarism.

The study guide refers students to specific sections of four programmed statistics texts. Students are encouraged to buy one of the programmed texts (Amos, Brown and Mink, 1965, Elzey, 1971, McCollough and VanAtta, 1963 or Smith, 1970), but are also encouraged to use other programmed texts if one is not sufficient. To make this practical for students, copies of each are kept on reserve at the college library.

In addition to the study guide and programmed texts, students have audio notebooks and transparencies to aid them in learning the materials of the course. Some of the tracks on the audio notebooks deal with mechanical matters such as the use of the calculators. Most of the audio notebook tracks are intended to be used with transparencies which illustrate statistical concepts. By overlaying transparencies, differences in distributions can be illustrated, curves can be compared and regression lines can be placed in scattergrams.

Programmed self instruction allows the superior students to finish the course without the aid of instructors or laboratory assistants before the end of the semester, thereby enabling instructional efforts to be expended as individualized aid for students having difficulty

mastering course materials. The statistics laboratory is open seventy-four hours per week with laboratory assistants present. It is equipped with 20 electronic calculators. Faculty instructors are available in the laboratory for six hours per week. Under these conditions, more highly individualized aid is given to students who need help than could normally have been provided under traditional classroom conditions, while at the same time, sparing students who are able to work independently on their own the possible boredom of traditional classroom lectures.

One major problem extant in the course before the application of computer technology was the tendency for students to plagiarize each others' homework. This was possible because each student's homework problems were based on data printed in the study guide. Since all students had the same set of data and the same homework assignments, each was expected to have the same answers. This led to students copying each others answers. While student cooperation among themselves can be pedagogically helpful, the mere copying of each others' answers was not. This defeated the purpose of a self study course. However, it was deemed desirable to preserve student cooperation. It was observed that often students could learn more from each other than from their instructors. But it was also desirable to prevent copying of each others' answers.

Individualizing a different set of data for each student's homework assignments, without the aid of a computer, would have required laboratory assistants to calculate each student's homework before correcting it, a very formidable task. Through the use of a computer, individualized data sets plus answers to homework problems could be easily generated, thereby eliminating the temptation of merely copying from another student, but at the same time, preserving the possibility of students cooperatively sharing a mutual learning experience.

The Use of the Computer in Developing Individualized Data Sets and Homework Answers

Initially it was planned to generate a separate set of data for each student. However, it was later decided that pedagogical advantages could be gained by establishing large population and then selecting student samples from the population. This provided the rare opportunity to show students the relationship between a population and a sample and to illustrate the concepts of sample variation, sample error, parameter and statistic.

The first problem in developing a population was the securing of data. The author had previously collected data from 1,000 Midwestern college students which provided data suitable for 5 of the variables students used in completion of homework assignments. Data for the remaining variables used in the homework assignments were not empirically available, so the computer was used to simulate data for seven more variables. Simulated variables were generated in such a way that certain population parameters would be insured. For instance, 30 percent of the population was urban, 55 percent was suburban, and 15 percent was rural. Females were generated with a slightly better (but statistically significant) High School Regents Scores and Freshman G.P.A.'s were simulated. Likewise, a relationship between place of birth and number of siblings was built into the data.

A punched card file of 1,000 data cards with the population data was produced and served as a data base from which random samples of 50 individuals were drawn for each student in the class. The data base was read on to a random access file and a random number generator was used to select samples of 50 cases from the file and wrote them on magnetic tape. The tape was used both for the printing out of each student's set of data on to individual data sheets for student use and as a data file for the computation of each student's homework assignment.

A program was written to print out the entire population card file so that it could be displayed on a tackboard in the Statistics Laboratory. The format was similar to that of the students' data sets so the students could readily see the source of their sample. This enabled the student to compare his sample to the population. To allow statistical comparisons, the population parameters were computed and posted.

Once the student data sets were established, it was only a matter of programming the homework assignments. The general philosophy in designing the programs that calculated the students' homework was to compute the statistics in the same manner in the computer as the students were using calculators in the laboratory. This made it possible to print out critical computations (i.e. means, sums of squares, expected frequencies, etc.) so that output could be used for diagnostic purposes as well as for correction of homework. A few departures were made from this approach. For instance, in tests of significance, sampling distributions were computed rather than looked up on tables.

Computer Costs for Development and Operation

The computer costs must be divided into two categories: first, the cost of development which includes machine time for drafting and developing programs, and second, the cost of operation which includes only the time for running programs which have been debugged.

The developmental costs of the FORTRAN programs which were required to do the eleven homework assignments as well as to generate and print sample data were \$109.75 on an I.B.M. 360/65. These were one time costs and are not incurred on subsequent utilization of the programs.

The operation costs are dependent on the number of students for whom data and answers are being generated. For 300 students, the cost of individualized materials was \$.36 per student. The greater the number of students, the less the cost per student. This includes the students' individualized data sheets and answers and intermediate computations for eleven homework assignments.

Machine Requirements

While an IBM 360/65 was used for this application, a much smaller machine could be used. The maximum core requirements for any program used on the 360/65 was 56K bytes or 14K words. The advantage of the 360/65 over many smaller machines was the ease of programming available in the extended 360 FORTRAN IV and the ease of creating and accessing data files.

The system of programs used a large tape file which contained the sample members for each student's data set. Tape facilities for such mass data storage are available on most college computer facilities. However, it would be possible to use a much smaller system without tape facilities, such as I.B.M. 1130, if one would be willing to give up the programming conveniences of an extended FORTRAN and use a more compact student sample storage techniques. Storage requirements could be reduced if only record numbers of student sample members were stored and student samples were reconstructed from the population each time it was needed by a homework program.

Student Evaluation of Course

A brief questionnaire was distributed among students so that student evaluation of the items were included as indicators of whether or not students' reactions to the course were favorable. Of approximately 300 students, 141 returned questionnaires.

One of the items used to ascertain overall approval of the course was the question, "Would you recommend this course to a friend?" An affirmative response indicated approval of the course and a negative response indicated disapproval. Sixty-two percent of the students indicated they would recommend the course to a friend, 35 percent indicated they would not. Two percent made no response (see Table 1). Clearly, a majority of the students approved of the course enough to recommend it to a friend.

	Number	Percent
Would recommend to a friend	88	62
Would not recommend to a friend	50	35
No response	3	2
Total	141	100

TABLE 1. Recommendation of Course to a Friend

However, approval of the course may not be a result of the techniques used (i.e., computer managed instruction and self instruction) but rather the subject matter itself. As a further indicator of approval of the instructional techniques, students were asked if they would like to have other courses taught in the same format. As indicated on Table 2, 61 percent answered affirmatively, while 37 percent answered they would not.

	Number	Percent
Would like other courses in same format	86	61
Would not like other courses in same format	52	37
No response	3	2
Total	141	100

TABLE 2. Preference for Other Courses in Same Format

Because of a concern that many students might feel isolated in a self taught computer managed course, an attempt was made to measure to what degree students thought the course was personal. While the potential for isolation exists, there also exists the opportunity for high personal involvement in the course which could offset feelings of isolation. To measure the degree of personalness expressed by students, a Likert type scale was given to students for their response on the evaluation questionnaire. Students were asked, "For you, did not having a regular instructor in a traditional class make the course impersonal?" Five responses were presented to students. They included, "Yes, very impersonal," "Only a little impersonal," "It was just as personal as my other courses," "Somewhat personal," "No, it was very personal." Twenty-three percent of the students responded that the course was very impersonal, another 26 percent indicated that it was only a little impersonal. Twenty-two percent indicated that the computer managed self study course was just as personal as their other courses. Nine percent indicated that it was somewhat personal and 18 percent indicated that the course was very personal. (See Table 3.)

Response	Number	Percent
Very impersonal	33	23
A little impersonal	37	26
As personal as other courses	31	22
Somewhat personal	13	9
Very personal	24	18
No response	3	2
Total	141	100

TABLE 3. Personalness of Course

Probably what does make the course personal for some students is the degree that they may become personally involved in the course and the degree to which they may pursue their own interests within the context of the course. While the data is not capable of supporting it, the writer suggests that the degree to which a course becomes personal is related to the degree to which a student can pursue his own interests through a choice of optional means for fulfilling the requirements of the course.

Many options could be built into the self instructional course. Optional homework assignments could be designed so that you would use discipline specific data. Students could

select from several homework assignments to fulfill the assignment for one unit. For example, in the test unit, students would be given the option to select income variables (Economics), academic difference variables (Education), mobility variables (Geography), or status variables (Sociology) to fulfill the homework assignment.

A computational option could be added to the course. Students now are required to complete the statistical computations on desk calculators. Several students indicated on the evaluation questionnaires that they knew how to use a computer and that use of the desk calculators was "busy work." For these students, the option of using a computer (desk top or larger) could be provided. To make this option a real choice, as well as providing the student with an individualized computer data printout, those that elect the computer option could be provided with a copy of his data punched in IBM cards.

To give the student a greater sense of control over the course, an interactive computational system could be employed to correct and grade students' homework. (See "Plans for Future Development.") In this manner, when a student was ready to have his homework checked, he could sit down at the teletype terminal and enter his answers. The computer would return corrected answers, diagnostic information, and record the student's performance in a computerized grade book. This would negate the necessity of turning in homework and waiting for a grader to correct it. No grading delays would prevent the student from progressing at his own pace and reinforcement would be nearly instantaneous. Moreover, personnel now involved in grading would be freed to give individual help to students in the statistics lab.

Plans for Future Development of Computer Management

In batch mode, the degree to which further individualization can continue is limited. Batch mode processing requires that all the homework be computed at once. Students must wait until laboratory assistants correct their work to proceed to the next assignment. In some cases, this impedes the student from progressing at his maximum speed. A partial solution to this problem would be to print out two copies of each set of homework answers and diagnostics. One set could be arranged for easy reference for the graders; a second set would be returned to the student with his corrected homework for diagnostics. Presently, the same set of answers are used for correction by graders and diagnosis by students. It is impossible to make this information available to students in the statistics laboratory before the homework assignments are due.

Such a system would still delay the evaluation of the students' work. Moreover, in batch mode, some types of homework problems cannot be handled. This is particularly true in those which the student is required to make a judgmental decision. For instance, in one homework assignment the student was expected to select his own class interval for a frequency distribution. It would have been impractical to try to anticipate every possible class interval a student might select and print out frequency distributions for each of them.

The best solution to handling the problem of quick correction of assignments and the evaluation of problems involving student decision, is to convert the approach to an interactive computing system rather than a system which operates in batch mode. Students could then get correction routines for each homework unit stored in the computer. In this manner, decision problems could take student decisions as input for parts of the correction procedure.

If an interactive computer system was used, a grade file could be established which would contain records of the students' homework performance. In short, a grade file in the computer would become the equivalent of an instructor's grade book. The grade file could be updated each time a student called a correction routine for his homework. Simple programs could be devised for instructors which would list student progress in terms of homework units attempted and performance in those units. Such information could be used by instructors to identify students who were having problems with the course so they could be singled out for individualized attention.

Such a system would require rigid security. Students would be tempted to try to receive credit for the course without doing the work. For instance, precautions would have to be taken to prevent a student from calling out a homework correction program, inputting answers that he knew were wrong, receiving correct answers and diagnosis, and then calling out the program again to input computer generated correct answers. Similarly, precautions would have to be taken to insure that students did not have "ringers" take the course for them or calculate their homework.

The establishment of grade files would also make possible the institution of individualized testing. An extensive pool of test items has already been developed. By bringing together grade files and a pool of questions over different areas of the course, an individualized final exam for each student could be developed which would cover the student's weak areas. Students could be told that they would only be tested over their mistakes in their homework. If they made no mistakes, they would receive no final exam and would only be graded on homework performance

(which would be perfect). If they wanted to know what they would be tested on, they could be told to keep their teletype printouts as records of their mistakes. From an instructional point of view, either way, the students learn the material.

In conclusion, the application of the computer has improved the quality of instruction in a programmed self instructional sample. The utilization of the instructional technique has been well received by students and does not seem to seriously degrade the students' sense of the personalness of the course. The continued development of this technique through the use of interactive computer terminals will enable the incorporation of optional ways the student may meet the course requirements thereby enabling him to tailor the course to meet his own needs and interests.

REFERENCES

1. Anos, Jimmy R., Foster Lloyd Brown and Oscar G. Mink. Statistical Concepts, A Basic Program. New York: Harper and Row, 1965.
2. Elzey, Freeman P. A Programmed Introduction to Statistics. Belmont, California: Brooks/Cole, 1971.
3. McCollaugh, Celeste and Loche VanAtta Statistical Concepts, A Program for Self Instruction. New York: McGraw-Hill, 1963.
4. Nasca, Donald J., Robert Potter and Frederick Halley A Guide to the Study of Statistics. Brockport, New York: Department of Educational Research, State University College at Brockport, 1971.
5. Smith, Milton G. A Simplified Guide to Statistics for Psychology and Education. New York: Holt, Rinehart and Winston, 1970.

THROUGH MULTIPLE REGRESSION AND THREE-WAY A M O Y
IN SOPHOMORE LEVEL APPLIED STATISTICS FOR THE BEHAVIORAL AND NATURAL SCIENCES:

Instructor-Student Demonstration of the Marchant Cogito 1016PR+IOTA-2

Clark I. Guilliama, Author
Robert Fletcher, Student
Missouri Southern State College
Joplin, Missouri 64801
Telephone: (417) 624-8100

Introduction

The object of this demonstration is to introduce instructors of applied statistics or experimental labs in the behavioral and natural sciences to the Marchant Cogito 1016PR+IOTA-2. The author's set for the presentation will be that of an instructor who teaches "Applied Statistics for the Behavioral and Natural Sciences" to sophomore level students. The student prerequisite is only high school algebra, or equivalent training. Emphasis is placed on the analysis and interpretation of student collected data (see Appendix I for the actual behavioral objectives presented the student on day one of course).

Description of the 1016PR+IOTA-2 or -3

As a manually operated calculator, the 1016PR combines the speed and quietness of electronics with the simplicity and easy-to-learn operation of a desk-top tape calculator. Accuracy is assured by factors and answers being printed on the tape with easily recognized symbols for their identification. The logical arrangement and markings of operating controls also contributes to optimum accuracy.

As a programmable calculator, the 1016PR offers some of the capabilities and the versatility of a small computer—all in the same compact, thirty-five pound desk-top unit. A program of up to 100 steps may be recorded, remembered, and automatically repeated. The unconditional and conditional branching features of the Cogito 1016 PR makes it attractive for programming courses also.

When the 1016 PR is combined with an optional IOTA-2 or -3 unit (Input-output Tape Accessory), programs can be recorded on magnetic tape and stored for later use.

The IOTA-2 makes it possible to store up to eleven 100-step programs in one cassette; and the IOTA-3, which we do not have, is basically the same as the -2, except one can dial the particular 100-step part of the tape he wishes. However, the time saved with the -3 is negligible; but it does facilitate identifying short programs on multi-step cassettes.

A library of small tape cartridges can be developed so that any frequently used program may be transferred from tape to the program memory of the 1016PR in just five seconds.

The 1016PR is algebraically correct in all mathematical operations. The SIGN Key permits the entry of negative numbers. All factors and results print with the correct sign. All answers automatically print with the correct decimal point if keyboard entries are indexed with a decimal.

After the Cogito prints an answer to any calculation, the result is always transferred back to the Keyboard Register. This provides a link between all the registers and eliminates any need to copy or re-enter immediate answers. Each of the six registers can contain up to 17 places, but only 14 digits will print. However, this is like 99 trillion plus.

Typical Course Problem

Suppose we are interested in examining the influence of three new and related drugs on the EEG activity of two diagnostic categories of rats (e.g., experimentally induced mania or depression). These data might well be cast in a two-by-three (2 x 3) matrix as in Table 1 below (the scores in each cell are individual EEG frequencies per second—highly improbably, but makes for dramatic results when interaction effects are to be emphasized).

DIAGNOSTIC CATEGORY	DRUG		
	1	2	3
I	8, 4, 0 $\bar{X} = 4$	10, 8, 6 $\bar{X} = 8$	8, 6, 4 $\bar{X} = 6$
II	14, 10, 6 $\bar{X} = 10$	4, 2, 0 $\bar{X} = 2$	15, 12, 9 $\bar{X} = 12$

TABLE I. Data for the two-way analysis of variance: $n=3$ EEG frequencies for two diagnostic categories of rats treated with three drugs.

532

This table illustrates a case in which main effect differences will not be manifested, but interaction does occur. A graph is a better way to depict the interacting effect (see Figure 1).

It is obvious that the two lines reflect "nonparallelness," but when compared with the within cell variance it may not prove to be significant, say, at the 0.05 level. Let us check, by approaching the data as if we had not graphed it and are students in a statistics lab presented only with Table 1 and access to the tape library (see Appendix 2) and the 1016PR-IOTA-2.

The student will decide on which cassette in the tape library is appropriate (demonstration begins). The operating directions are on each cassette, as well as the printout (the company provides several standard programs which the instructor can "dump" on tape, but the instructor can often modify a tape(s) for greater efficiency). The student should pick the cassette labeled "AOV 2-WAY, -M's." The directions read:

1. Reset, Clear All.
2. Set Conditions: Round Off up; Select Print down, Decimal as you please, IOTA on Mult.
3. RECORD (1st 100 steps of program are entered).
4.
 - a. Enter data by row, Run after each score.
 - b. After each row completed, Run.
 - c. After all rows completed, Run.

Prints: $\sum_{r=1}^R T_r^2$ (sum of the squared row totals).
5. Enter RC (no. in cell x no. of col's), Run.

Prints: $1/n \sum_{r=1}^R T_r^2$

Prints: T (or $\sum_{i=1}^N X_i$).
6. Enter N (or nRC).

Prints: T^2/N [or $(\sum X)^2/N$].
7. Record (records 2nd 100 steps of program).
8.
 - a. Enter data by columns, Run after each score.
 - b. After each col. completed, Run.
 - c. After all col's completed, Run.

Prints: $\sum_{c=1}^C T_c^2$ (sum of the squared col. totals).
9. Enter RC (no. in cells x no. of Rows), Run.

Prints: $1/n \sum_{c=1}^C T_c^2$
10.
 - a. Enter data by cell, Run after each score.
 - b. Run after each cell.
 - c. After all cells completed, Run.

Prints: $\sum_{RC} T_{RC}^2$ (sum of the squared cell values.)
11. Enter n (cell no.)

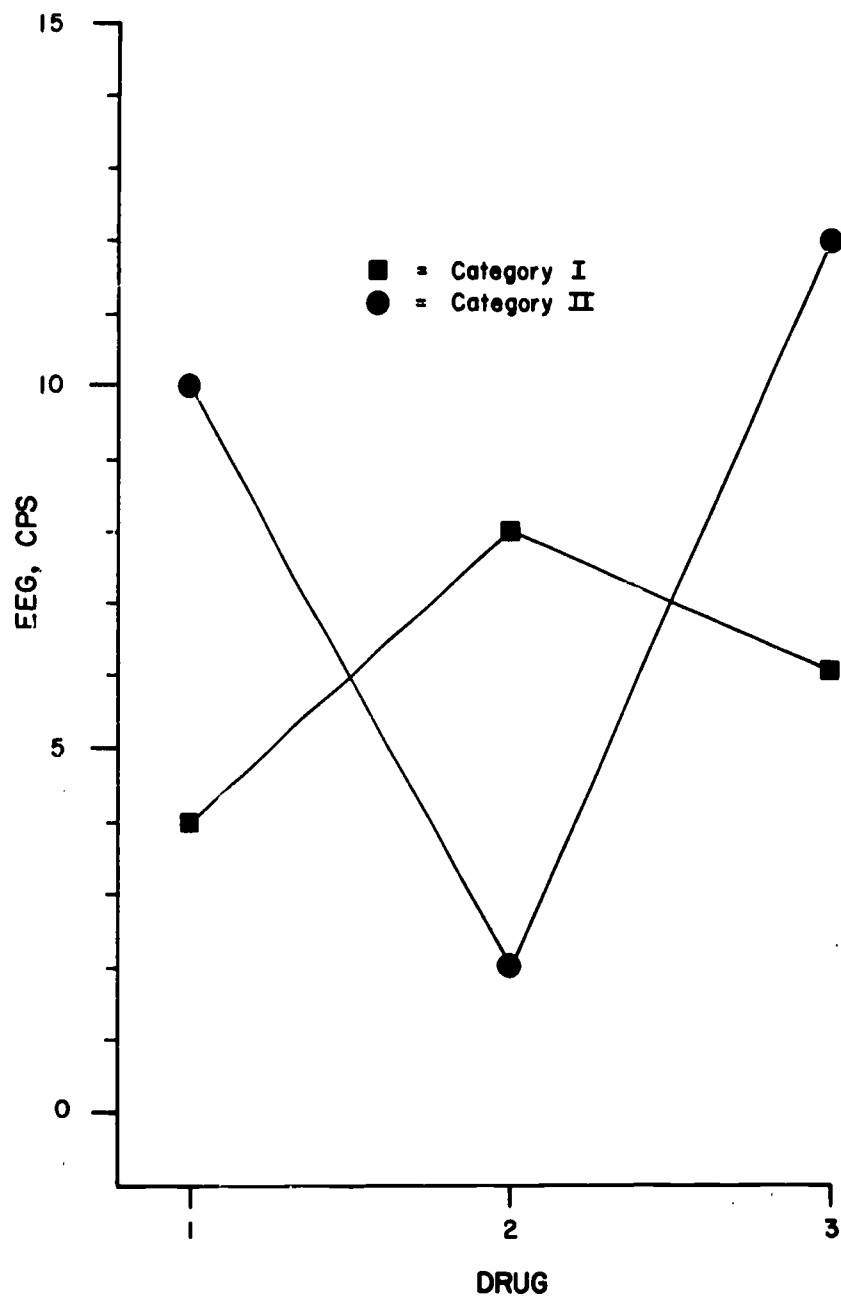


FIGURE 1. Mean EEG scores by drug and diagnostic category.

Prints: $1/n \sum_{RC} T^2$

12. Record (3rd 100 steps of program are entered).

13. Run.

Prints: $\sum_{RCI} X^2 - 1/n \sum_{RC} T^2$ (sum of squares within cells).

14. Enter: N-RC (df for within cells sum of squares).

Prints: S_{w2} (variance within cells).

Line spaces.

Prints: $1/n \sum_{R} C T^2 - T^2/n$ (sum of square Rows).

15. Enter: R-1 (df Rows).

Prints: S^2_R (variance between rows) and F_R

Line spaces.

Prints: $1/n \sum_{C} R T^2 - T^2/n$ (sum of squares between col's).

16. Enter: C-1 (df col's).

Prints: S^2_C (variance between col's) and F_C

Prints: $1/n \sum_{RC} T^2 - 1/n \sum_{R} C T^2 - 1/n \sum_{C} R T^2 + T^2/n$ (sum of squares interaction)

17. Enter: (R-1) (C-1) (or df interaction).

Prints: S^2_I (interaction variance) and F_I .

Prints: $\sum_{i=1}^N X^2 - T^2/n$ (or sum of squares total).

STOPS!

The student rips off printout and notes the F ratios. The required $F_{.05}$ values for rows, columns, and interaction, respectively, are 4.75, 3.88, and 3.88. In this particular problem, only the obtained F ratio for interaction (8.1509) is significant at the .05 level. F_R is 2.0377 and F_C , 2.7169.

Because of the limited time neither a three-way ANOV nor a multiple linear regression analysis will be demonstrated but the cassettes are in our tape library (see Appendix 2). We would be willing to duplicate our tapes if interested persons will send us blank tapes and specific programs needed.

Evaluation

Before Missouri Southern State College started using the programmable printing calculators (desk computer?) the instructor managed to get the sophomores through only simple linear and nonlinear regression, and one-way ANOV (even with electric calculators). However, our students now complete three-way ANOV and multiple regression (up to four variables). The quality and sophistication of the student papers in later lab courses and independent study has increased significantly as evidenced by joint publications of staff and students. The student demonstrator for the program above is a case in point as he has a two-way ANOV in press with The Journal of Biological Psychology. The "little computer," "in-between calculator" or "desk computer" has greatly reduced the student's real or supposed inability to handle calculations and permits the emphasis to be placed on more appropriate behavior; viz., mastering three or four basic concepts that are used over and over again (e.g., sum of squares), the logical mechanics of statistical

thinking, and some rather peculiar semantic problems arising from statistical jargon that inhibit a clear-cut decision-making process.

If one has huge N's and more than four or five variables, the obvious route is the punched card; but there are so many studies that students and faculty could do--and need to be doing--that are in-between calculators and big computers. So, reduce your anxiety about procrastinating or computer time and invest in a relatively inexpensive electric programmable printing calculator--thirty-five pounds of gold.

APPENDIX I

Behavioral Objectives for Psy 311 Applied Statistics for the Behavioral and Natural Sciences Prepared by Dr. Guillians

Prologue

The overall objective of this course is to introduce students and research workers in psychology and education to the concepts and applications of statistics. Emphasis is placed on the analysis and interpretation of data resulting from the conduct of experiments. Students and investigators in natural or pre-life science programs, sociology, and other disciplines may also find the course useful.

The course is designed to permit the instructor some freedom of choice in the selection of material to be covered (as a function of past experiences of the individuals enrolled).

Assumptions. Each student can demonstrate prerequisite skill in high school or freshman (college) algebra, or the equivalent training.

Specific Unit Behavioral Objectives.

Unit I

The student will: experience a need for knowledge about statistics by reviewing several current books, journal articles, and research bulletins in an area relevant to his professional training program and noting the terms from the result sections he does not comprehend.

Unit II

The student will: deduce--by intuition, trial-and-error, and instructor reinforcement strategies--formulae for simple descriptive statistics; viz., for measurements of central tendency, variability, and standard error.

demonstrate skill in operating an electric calculator and/or an electric printing-programmable calculator in solving raw score applications with the formulae above, and encountered throughout the remainder of the course.

Unit III

The student will: analyze functions of linear and nonlinear curves; viz., the normal curve.

Unit IV

The student will: derive--by intuition, trial-and-error, and instructor reinforcement strategies--formulae for simple regression and simple correlation problems.

plot scattergrams for bivariate data and draw, by intuition, lines which best fit the data (expected regression lines); also draw the regression lines after solving for the regression coefficients (beta and alpha, or the slope of the line and the intercept of the criterion variable).

calculate errors of measurement and errors made in estimating population values from sample values; viz., the standard error of: the mean, measurement, and estimate, Calculate Eta, the correlation ratio, and its unbiased estimate, Epsilon.

Unit V

The student will: calculate tests of significance for means; viz., z, t, and F (one-way).

translate his obtained z's, t's and F's into meaningful verbal statements relevant to decision-making.

calculate tests of significance for nonparametric distributions.

Unit VI (presented on independent study basis, or to entire class, if time and past achievement records warrant it).

The student will: solve multiple and partial regression equations with linear and/or non-linear terms.

translate his regression coefficients into meaningful statements relevant to decision-making.

solve F ratios for two and three-way ANOV problems.

Epilogue

When the course is completed the student SHOULD have an intellectual grasp of the practical technology of statistics, and the communication skills necessary to transform statistics into meaningful statements relevant to decision-making.

APPENDIX 2

Department of Psychology Tape Library
Missouri Southern State College

1. Mean, standard deviation, standard error of mean (ungrouped data).
- 2,3. a. Summations: $N, \sum X, \sum X^2, \sum XY, \sum Y, \sum Y^2$; simple linear regression: b_{yx}, a_{yx}, r, S_y (raw score entries).
b. Linear regression: N, b_{yx}, a_{yx} and Y' as a function of any given X
c. X, S_x, Y, S_y , and r .
4. Multiple correlations.
5. Correlation: N and r (raw score entries)
6. Chi-square (varying theoretical frequencies)
7. Chi-square (constant theoretical frequencies)
8. a. t statistic (must enter N_1, X_1, S_1 , and N_2, X_2, S_2).
b. t statistic for correlated samples (enter X - Y raw score pairs)
c. t statistic for independent samples (raw score-ungrouped data entry).
9. a. One-way ANOV (raw score-equal n 's): sums of squares, variances, and F ratio).
b. One-way ANOV (raw score-equal n 's): $\sum X_j$ and n_j for each column, $N, \sum x_B^2, S_B^2, \sum x_W^2, S_W^2, F$.
10. Two-way ANOV (raw score-equal n 's): row, column, and cell sums, n 's and means, respectively; sums of squares, variances and F ratios.
11. Three way ANOV (raw scores-equal n 's): ditto tape #10 plus main effect for layers and additional interaction F ratios.

CONDUIT - A Concrete Pipeline for Software-Starved Little People

Joseph E. Deak
North Carolina Educational Computing Service
Research Triangle Park, North Carolina 27709
Telephone: (919) 549-8291

Major: All the roles of the computer will enhance curricula.
Minor: Canned programs represent one role of the computer.
Conclusion: Canned programs will enhance curricula.

INTRODUCTION

This syllogism should not be lightly regarded or cast off as simplistic. Countless millions of dollars have been spent in full belief that both premises are valid. The Pierce Report[1] (forgive the reference to overstressed antiquity) implied the major premise throughout in its rosy picture of the ultimate effect of the computer on education. Almost everyone believes the minor as witnessed by the contributions to national conferences[2,3,4]. In several isolated situations, but by no means on a grand national scale, the conclusion has been qualitatively justified. The failure of canned programs to accelerate computer-based curriculum development is apparent despite NSF funding of 25 networks many of which sponsored the support of canned programs as curriculum development, despite several efforts at establishing software exchange centers, and despite funding by ARPA, NIH, OE, and other support agencies.

There exists no quantitative proof that canned programs are indeed potential enhancers of curricula. Oh, there are some indicators and good ones which will be referenced below. Before these indicators are worth pursuing, a critical missing link must be supplied. Canned programs must be transportable for the "LITTLE PEOPLE" who cannot benefit from such materials unless they are available.

CONDUIT, Computers at Oregon State, North Carolina Educational Computing Service, Dartmouth, and the Universities of Iowa and Texas, represents a practical solution to the transportability problem involved in exchanging software. This current cooperative national project has been structured to test transportability of selected canned programs to 100 colleges involved in 5 regional networks. Hundreds of canned programs for computer-based undergraduate curricula in 11 disciplines have been or are being prepared for exchange and usage by the networks which include many "Little People." Surveys of the transportability problem, the present status of program exchange, and compelling examples of canned programs in CONDUIT will be attempted.

TRANSPORTABILITY - THE PROBLEM AREA

The difficulties involved in getting canned programs to the "Little People" (faculty without big computer centers or staffs) become apparent only when you start working in program exchange[5]. Requests for programs from your exchange centers are numerous (NCECS receives about 500 a year from outside the network) indicating further that canned programs may be valid educational materials. Moving these materials effectively has not come about despite some 60 historical attempts at setting up centers[6]. Four categories of deficiencies in producing effective transportability can be discussed for the parameters involved: absence of documentation, technical problems, misunderstanding of transportability, and weaknesses in exchange centers.

Absence of Documentation: There is not a sufficient reward structure for a guarantee of the production of serious documentation. Educational publications rarely pay off for tenure in the traditional disciplines (outside of education itself) except in small colleges where the push for research is not intense. (NCECS received over half of its program packages from such institutions[7].) Publishers are also wary of applications software since these materials seldom bring profit (ask IBM) and they pass around underground to the point where the programs themselves are almost without value.

Even where documentation is available, it is non-standard; it has insufficient content as to the meaning of the program, the educational technology necessary to use the program, the possible educational validity of the package, and the educational philosophy behind the concept. While there are exceptions to these generalizations, these deficiencies are not new to anyone. But documentation is not nearly enough.

Technical Problems: Transporting a 100 line FORTRAN Program is a trivial matter and usually produces trivia on the receiving end. While 100 line programs can be exciting, the programs

which usually produce educational innovation are much more complex and can be classified as systems or packages. Machine differences, language translation, file organization, and storage capacity are among the technical problems facing the importer. Coupled with the absence of documentation on these technical problems is an absence of a competent technical staff to implement the system from the foggy matter received as an educational tool.

Surprisingly, these technical problems are not serious for transportability! They are easy to overcome utilizing relatively inexpensive technical personnel if the concepts to be transported are known. Transportability is misunderstood, but without any fault being imputed to the "Little People."

Misunderstanding of Transportability: An extremely complicated system can be transported without using the data bases, programs, or even the algorithms of the original system[8]. The real value of computer-based educational packages lies in the philosophy, pedagogy, and substance of the package. It is often far easier to use a good programmer to reconstruct a system once these essentials are available than to attempt to modify an imported program or system.

But philosophy, pedagogy, and substance are rarely considered in the transportability process. Educational philosophy is primitive in most of the available canned programs. Pedagogy is usually absent to say nothing of the rarity of student involvement in the design of the pedagogy. Dangerous "black boxes" are abundant. Once a package is transported, the relevant documentation of the process is rarely, if ever, available, making repetition of transportation necessary. Computer center philosophies and built-in ignorance of the needs of other disciplines are no help. Current or past exchange centers have deficiencies of their own.

Exchange Center Deficiencies: Exchange centers have provided one good data point - People ("Little and Big") want programs. What they have not provided is efficient transportation of programs. At least until this writing, they also have not been able to support themselves.

EIN[9,6] was established to foster program usage where the program was running and by users with no terminal link to that computer. Only two usages of 150 possible programs were reported after a year of activity and these were set up to test the process. LeGates reported hundreds of exchanges occurred underground despite no formal structure for this purpose.

The Quantum Chemistry Program Exchange, QCPE, has distributed many sophisticated programs, packages, or systems, but admits these are mainly research and not educational applications[10]. As a result, EMU-CECCP (Eastern Michigan University - Center for Exchange of Chemistry Computer Programs) was set up to fill this void[11]. This center offers very good documentation on what meets its standards, receives hundreds of requests, and may reach excellent maturity before its financial basis collapses. However, insufficient conceptual communication as to educational innovation is available for most of its holdings (a problem of resources and not the center itself). A large number of the concepts are readily reproducible without bothering with the programs themselves. No actual implementation of programs is available from this center.

Several of the regional networks have established pseudo exchange centers for the materials developed or imported during their orgies of curriculum development. The first one historically was the COPES library of the Illinois Institute of Technology network[12]. About 400 contributions ranging from programs to teaching units provided such groundwork but few of these reached any wide acceptance. PALS of NCECS[7] was a larger compilation but is so network-related as a current awareness system that it offers little toward solving the complexity of the needs of the "Little People" outside of the North Carolina region.

Beloit College's SSIPP[13] provided one of the most effective centers of concern for the Little People. Its 500 or so programs, written in an interactive version of FORTRAN, were well documented with significant aids provided for translation to other systems. The Center was well managed and requests were efficiently fulfilled. Although good conceptual communication was involved in a large number of these programs, the social science subset is not large, nor is there evidence of educational value in the programs dealing with social science and not strictly statistics. It would be interesting to find out how many centers implemented programs with SSIPP since the transportability problems involved are non-trivial. (NCECS has the entire library and has implemented only 4.)

DECUS (DEC Users Library) has reported that the exchange center function seems to be a waste of time[9] such to the amazement of this writer who violently disagrees. ENTELEK suffers from representing only a subset of computer-based educational materials. COSMIC is too costly. The list goes on. "Little People" can rarely use these centers.

Catalogs are an abundant source of materials further indicating their availability. Many centers have published a spectrum of such compilations ranging from mimeographed handouts like PALS[7] to ambitious formal efforts like that of the Joint Users Group (JUG) of ACH[14]. While these compilations are of tremendous value, they do not meet transportability head-on. "Little

People" end up without such hope of getting their hands on the materials in operational form or even of transporting the materials effectively.

Not that there is a lack of interest! Despite some attitudes of computer center people (Carnegie-Mellon's computer center philosophy: if the documentation and systems available are not understood by the user, that's his problem[9,5]) and attempts to limit development under the financial squeeze (Kentucky and Oklahoma Universities' computer center policies on control of educational usage[15]) some organization has occurred to explore the transportability problem. The Association for Computing Machinery, ACM, has two special interest groups, SIGSOC and SIGCUE, with task forces dedicated to the study[16,17]. CONDUIT, the five regional network cooperative, is the most recent specific effort and was funded by the National Science Foundation[18]. The work begun in transportability in the regional centers and the goals of CONDUIT should provide the reader with some concrete solutions to the abstractions above.

TRANSPORTABILITY - SOME SOLUTIONS

Transportability has been accomplished by uncounted hundreds. "Little People" and others have not only moved canned programs but have produced a wealth of material. In the regional centers, a great deal of this work has been going on in many disciplines. The foundation of CONDUIT can be specifically illustrated in several disciplines. It must be understood that the following specific cases have a built-in sifting process into which conventional, or somewhat prejudiciously unclear examples have emerged.

Accounting: Wilbur Pillsbury of Kean College produced a significant package of programs[19] (FORTRAN IV) which augment two semesters of elementary accounting with an apparent removal of drudgery. This package has been refined through summer workshops for faculty of many institutions and, according to Pillsbury, has drastically reduced the flunk-out rate of first-year accounting students.

Transportability of the Pillsbury system (COMBUGUIDES ONE and TWO) has been greatly enhanced by the involvement of the publisher of the materials[20]. The package is available and clean for IBM and CDC equipment. Further, Pillsbury has actively involved himself in the training process at various sites so that the philosophy, pedagogy, substance, and limitations of the package are transported with the system itself. Actual implementation of the package at any one site is relatively simple but not without some of this professional guidance. Of the CONDUIT members, both the Iowa and North Carolina networks have implemented this package.

The KOONTZ package[21] (PL/I) of North Carolina is an example of several programs available for accounting education that have not been made easily transportable. Missing are some essential elements for transportability: philosophy, documentation, pedagogy, and implementation methodology.

Biology: The Iowa network produced 30 modules for teaching biology with the computer as a tool[22] as well as an integrated course in mathematical biology. (Modules or teaching units are really an attempt to wrap up the program with the philosophy, pedagogy, and theory involved.) Of the 30 made available, 5 have found acceptance by a biologist in North Carolina who redid all 5 programs. What is transportable? Surely not the programs.

Business-Economics: A wide variety of models in both Business Administration and Economics have been cataloged as available[23]. Publishers have pushed a wealth of these packages without any solid indications of very wide acceptance or implementation for single modules. Transportability has ranged from extremely difficult (usually for the sophisticated models) down to trivial (usually for the simpler models). To the "Little People", the problems of importing a game are complicated by a lack of knowledge as to how good the educational experience may be, as to how to manage the model, as to the complexity of the package, and as to the educational level required for its use.

The Executive Game[24], TEXG, represents an easy-to-import management game that can be used in early undergraduate education. Because the program was well-tested at the University of Michigan as well as carefully documented and distributed by the publisher, TEXG easily produced wide acceptance in the North Carolina network. Built-in game management assistance and criteria for team evaluation help make the game comfortable for the "Little People" (several community colleges have adapted TEXG). A state-wide intercollegiate competition in TEXG run by the North Carolina network was an important factor for transportability.

The "International Operations Simulation[25]," INTOP, represents a sophisticated game without any of the transportability ease found in TEXG. Experienced faculty users, intricate management, limited usage (advanced students), and difficult implementation problems are involved for getting this to the "Little People." The North Carolina network has put in 6 months of activity just getting ready to offer this game on the level of intercollegiate competition.

Between these games lie a wide variety of available models whose transportability is complicated by all the factors mentioned in describing TEXG and INTOP. A history of the movement of any model is something unpublished and the missing base of philosophy, systems requirements (and cleanliness of the package), pedagogy, etc., make the games almost untouchable by the "Little People." The North Carolina network has tried several packages and needs to communicate the results of these tests. CONDUIT is planned for this purpose.

Chemistry: Exchanges for chemistry programs[7,10,11,12] have made hundreds of educational packages available without affording much data on transportability. While the majority of these are of the mobile 100-line FORTRAN variety, sophisticated systems remain relatively immobile. The tutorial monsters of Illinois' PLATO[26] and Texas' system[27] have not moved nor has the North Carolina infrared spectral searching system, ISIS[28].

But the ISIS system of North Carolina was an import! The American Society for Testing and Materials, ASTM, made the huge data base available but the available searching systems were cost prohibitive to education. ISIS was built with an educational philosophy in mind and only the data base was needed--an indication of the all-but-proven fact that transportability doesn't necessarily involve programs.

Mathematics: CRICISAM, Center for Research in College Instruction of Science and Mathematics, and its computer-based calculus book[29] is just too much. Clearly, mathematicians don't like canned programs either. No single compelling example of acceptance of a canned program can be given by the North Carolina network. Others may change this story, but project COEXIST at Dartmouth College[30] is progressing with the concept that canned programs are relatively useless in mathematics, physics, and engineering curricula.

Physics: The 11 programs published by the Commission on College Physics[31] have been transported to an uncounted number of centers, all 11 having been implemented in the North Carolina network. These programs are better than the "100 line" variety yet provide problems in transportability. Sample input is missing. Systems limitations are not treated (probably couldn't have been) and there are bugs. But these are standard problems. The concepts were very well documented and these programs afford decent transportability. Not very many others do.

The Commission on College Physics' conference in 1970[32] offered several canned programs to the user, the response to which was surveyed by the HUMPRO[33] corporation and is yet to be published. Early indications are that over one-third of respondents went after these programs. This author did not find the canned program resource in COMUSE as rich as the conceptual benefits. Most of the concepts presented could only be imported without programs.

Social Sciences: The most published undergraduate computer-based system is IMPRESS of Dartmouth College[34-37], a survey analysis system. POISSON[3], Package of Instructional Social Surveys of North Carolina, is the first daughter of IMPRESS, a feat of transportability to IBM time-sharing from the Dartmouth system that took one year. Remarkable in this process is the fact that NO PROGRAMS, NO ALGORITHMS, and NO DATA BASES were imported. POISSON looks just like IMPRESS but was rebuilt from concepts and output alone. Transportability was implemented by importing philosophy (two seminars in North Carolina by Dartmouth people), pedagogy (the tutorial content was adopted as much as possible), documentation (codebooks and primer), systems requirements (flexibility of the system itself), and substance (a textbook[38]).

HUMANS, Homans' Model of Human Interaction, was developed in the IIT network, and transported to the North Carolina network. The model itself was easily transportable but offered nothing until a small group was simulated and the result presented to a workshop of social scientists. Wide adoption and subsequent development resulted mainly from the addition of conceptual content to the model itself.

CONDUIT - A TEST OF TRANSPORTABILITY

CONDUIT was built to concretize the process of transporting computer-based educational materials so that they become available even to the "Little People." From the transportability problems sketched above (mainly for the North Carolina node of CONDUIT) it can be seen that the process cannot be boiled down to a transfer of programs with documentation. Machine-independent languages, translating packages, and super-systems are the computer scientists' answers to this problem and do not effect the transfer of concepts, pedagogy, philosophy, etc. The Rand Report[39] touched on the problem areas by indicating that the lack of an effective program dissemination mechanism (coupled with the lacks in standardization of facilities and documentation) was a serious impediment to the growth of usage of canned programs. CONDUIT was designed to transport about 100 packages taking into consideration all of the parameters deemed necessary to effectively move concepts, pedagogy, philosophy, and especially substantive content of the materials themselves.

CONDUIT is structured as a small central organization located in North Carolina and directing the design of the transportability mechanism through the five regional networks around Dartmouth College, North Carolina Educational Computing Service, Oregon State University, and the Universities of Iowa and Texas. Each regional network has a local coordinator for CONDUIT with programming assistance. The local coordinator is responsible not only to the CONDUIT function but also to his network of users which is composed of institutions of all sizes. Since the transportability process must involve discipline-oriented faculty as well as computer scientists, CONDUIT provides a large pool of such a mix, many individuals of which have already been involved in the process itself.

Several functions should precede the ultimate design of an effective mechanism for transportability and these are implied in the purposes of CONDUIT:

- (1) To more effectively promote program exchange.
- (2) To eliminate duplication of efforts in transportability.
- (3) To offer the nation a catalog of materials that include data on transportability.
- (4) To sift out the more relevant educational applications.
- (5) To test the cost of using selected materials in the classroom.
- (6) To provide data on the need for revision and on the flexibility of the packages.
- (7) To provide a test on the effectiveness of transportability toward curriculum development.

From these purposes it can be seen that CONDUIT is really a test of transportability. The 100 or so packages to be transported will definitely be available to all the networks but they will represent a subset of what is available. A test should, however, add to the definition of a working mechanism.

Three very different configurations are involved—CDC 6600 environments at Texas and Oregon State, IBM360 and 370 systems at Iowa and NCECS, and the HIS system at Dartmouth. Transportability between CDC users and IBM users will be accomplished along with the process between the three environments. The coordination of these efforts should elucidate and establish criteria for communication for all environments.

Educational validity is not to be a testing subject for CONDUIT. The validity of canned programs for education is currently being studied by HUMERO[40]. CONDUIT specifically deals with the process of transportability and the test of the effectiveness of this process will be evaluated by an outside agency.

The fundamental differences between CONDUIT and preceding efforts at program exchange are noteworthy. CONDUIT will actually move materials and support the process. The design of the test and the process itself will bring together several representatives of each discipline representing a spectrum of configurations and institutional sizes. Since 11 disciplines will be involved (accounting, biology, business administration, chemistry, economics, geography, mathematics, physics, political science, sociology, and statistics), the interchange between discipline groups should expose several mutual areas of interest and eliminate duplication of the same things in several disciplines. The catalog of CONDUIT will reflect transportability and not just the big unknowns of availability. The presence of a central organization coordinating the efforts of the various environments guarantees the focus on importing rather than merely offering what is available locally.

CONDUIT was funded[41] November 1, 1971, and the central organization is directed by Ronald Blum, formerly of the Commission on College Physics. The networks have produced hundreds of applications ready for exporting and several have already been moved. Canned programs will be available to the "Little People" whose acceptance and subsequent usage will provide indicators of the validity of canned programs in education. Since the author is a firm believer and witness of the truth in the syllogism (at least in North Carolina) upon which we embarked, CONDUIT is predicted to say—"I told you so!" This prediction is based on the faculty and students who are now or will be involved in transportability itself—a process that rarely has been considered for any educational package beyond implementing it locally.

REFERENCES

1. President's Science Advisory Committee, "Computers in Higher Education," U. S. Government Printing Office, February, 1967.
2. Proceedings of the Conference on Computers in Chemical Education and Research, Northern Illinois University, DeKalb, Illinois, July 19-23, 1971.
3. Proceedings of the Second Conference on Computers in the Undergraduate Curricula, Dartmouth College, Hanover, New Hampshire, June 23-25, 1971.

4. Proceedings of the First Conference on Computers in the Undergraduate Curricula, University of Iowa, Iowa City, Iowa, June 16-18, 1970.
5. J. R. Denk, "Exchange of Applications Programs for Education--A National Statement," *INTERFACE* 5, pp. 11-21, February, 1971.
6. J. LeGates, Private communication. Mr. LeGates was director of EIN, Educational Information Network, a branch of EDUCOM dedicated to testing program usage at centers where programs were operational by users not "wired" to these centers.
7. PALS, Program and Literature Service. A twice yearly catalog-journal of NCECS for current awareness of available canned programs. PALS-1971-01, the sixth issue, is available from the author.
8. J. R. Denk, "POISSON--A Daughter of Dartmouth's IMPRESS Has Been Born in the Environment of IBM Time-Sharing," Proceedings of 1972 Conference on Computers in Undergraduate Curricula, Southern Regional Education Board, Atlanta, Georgia, June 12-14, 1972.
9. Author's view of panel discussion involving the program exchange taskforce of EDUCOM, Atlanta Meeting, Fall, 1970.
10. H. Shull, "National Cooperation in Theoretical Chemistry Program Exchange and Possibilities for a National Laboratory," reference 2 above, pp. 7.2-7.4
11. R. W. Collins, "The Eastern Michigan University Center for the Exchange of Chemistry Computer Programs, EMU-CECCP," reference 2 above, pp. 7.33-7.38.
12. Cooperative Program Exchange Service (A Library of Programs and Teaching Units for 11 Disciplines); Cooperative Venture in College Curriculum Development, Illinois Institute of Technology, Chicago, Illinois 60616.
13. Social Sciences Instructional Programming Project, Beloit College, Beloit, Wisconsin 53511.
14. "Computer Programs Directory 1971," Edited by Ben R. Faden, CCM Information Corporation, Subsidiary of Crowell Collier and Macmillan, Inc., New York, 1971.
15. Panel Discussion, Spring Joint Computer Conference, Atlantic City, New Jersey, May 18-21, 1971.
16. SIGSOC, Special Interest Group for the Social Sciences; taskforce chaired by R. W. Anderson, Professor Sociology, University of Minnesota.
17. SIGCUE, Special Interest Group for Computers in Undergraduate Education; taskforce chaired by J. R. Denk of NCECS.
18. Grants GJ-31752 through GJ-31757, December 2, 1971.
19. W. F. Pillsbury, "Computer-Augmented Accounting Education at Knox College," reference 3 above, pp. 516-522.
20. W. F. Pillsbury, "Computer-Augmented Accounting--Compuguides One and Two," South-Western Publishing Company, Cincinnati, Ohio, 1970.
21. C. C. Koontz, "Computers and Computer-Augmented Accounting at Lenoir Rhyne College," reference 3 above, pp. 508-515.
22. For information on these units, write G. P. Weeg, Director of the Computer Center, University of Iowa, Iowa City, Iowa. These units were published in the summer of 1971.
23. R. G. Graham and C. F. Gray, "Business Games Handbook," American Management Association, Inc., 1969, Oregon State University.
24. J. R. Jackson and R. C. Henshaw, Jr., "The Executive Game," Richard D. Irwin, Inc., Homewood, Illinois, 1968.
25. H. B. Thorelli, R. L. Graves, L. T. Howells, "International Operations Simulation," Free Press, New York, 1963.
26. S. Smith, "Instruction in Chemistry Using Plato," reference 2 above, pp. 4.39-4.46.

27. S. J. Castleberry, "The Development of Computer Based Instructional Systems in Chemistry," reference 2 above, pp. 4.2-4.14.
28. J. B. Denk, "ISIS - A Live Infra-Red Data System for Chemistry Education," reference 3 above, pp. 347-358.
29. W. B. Stenberg and R. J. Walker, "Calculus, A Computer-Oriented Presentation," CRICISAN, Florida State University, 1969.
30. Computer-Based Course Materials for Introductory University Mathematics, Physics, and Engineering--a writing project in its second year at Dartmouth College.
31. "Computer-Based Physics: An Anthology," Edited by Ronald Blum, Commission of College Physics, September, 1969.
32. COMUSE: Computers in Undergraduate Science Education, Proceedings published by the Commission on College Physics, August 17-21, 1970, Illinois Institute of Technology, Chicago, Illinois.
33. HUMRRO, Human Resources Research Organization, 300 North Washington Street, Alexandria, Virginia. Private communication from B. Hunter of HUMRRO.
34. Interdisciplinary Machine Processing for Research and Education in the Social Sciences. E. D. Meyers, "An Introduction to Project IMPRESS," Time-Sharing Colloquium, Kiewit Computation Center, Dartmouth College, February 19, 1970.
35. J. A. Davis, "Using the IMPRESS System to Teach Sociology," reference 3 above, pp. 382-388.
36. E. D. Meyers, "We Don't Know What We're Doing," reference 3 above, pp. 159-170.
37. E. D. Meyers, "IMPRESS and Undergraduate Education in the Social Sciences," reference 4 above, pp. 8.23-8.29.
38. J. A. Davis, "Elementary Survey Analysis," Prentice-Hall (Methods of Social Science Series), Englewood Cliffs, New Jersey, 1971.
39. R. F. Levien et al, "The Emerging Technology," Rand Corporation, Santa Monica, California, Preliminary Draft, September, 1970, p. 111.
40. R. J. Seidel, B. Hunter, and M. L. Rubin, "A Study of Computer-Based Curricula: The Process of Change," a study brief, Human Resources Research Organization, 300 N. Washington Street, Alexandria, Virginia, November, 1971.
41. "CONDUIT, A Proposed Solution for the Implementation and Evaluation of Curriculum Development via Dissemination of Computer-Based Materials," July 15, 1971. Proposal's principal author was J. B. Denk. Copies may be requested from author.

AN EXPERIMENT IN COMPUTER TRAINING FOR COLLEGE FACULTY

Ronald L. Code
Stanford University
Stanford, California 94305
Telephone: (415) 321-2300

The focus of this year's conference, like that of the two preceding ones, is on the innovational ways in which computers are being applied to the undergraduate curriculum. Most of the papers presented assume that the instructor has at least enough knowledge of the computer to enable him to guide his students and answer any fundamental questions about computer techniques. Unfortunately, most college teachers are not versed in computer programming and many have only a vague notion of the computer as a general tool. This is one of the major reasons for the slow adoption of computer oriented materials into the classroom. Most teachers have received no formal computer training themselves, are hesitant to change proven methods, and are understandably reluctant to place themselves in a position which may expose this gap in their knowledge. Thus, the faculties of colleges, in general, can be characterized as having a few computer enthusiasts and many computer "agnostics."

The National Science Foundation recognized this problem, and within the last two years has funded several computer consortia which have as their function the training of college faculty[1]. This is a very different approach than was used in earlier projects. Most of these early "networks" simply provided "raw" computer power to a large number of schools. This was usually accomplished by a large university supplying a timesharing or remote batch service to several smaller schools. It became apparent that more of a seed effect would be accomplished by directing attention to the faculty rather than to the students and the existing demand.

In April of 1971 the Stanford Computation Center received a grant[2] to form a regional computing network which would be served by Stanford's IBM 360/67 and its existing services[3]. The Northern California Regional Computer Network is comprised of eighteen colleges and universities[4] which are representative of centers of higher education in the United States. Half are junior colleges while the others offer four year and advanced degrees. Eight are privately funded and ten are part of a state, county or city system. Some are located in major population centers, others in suburban settings, and some in remote locations. Student bodies with as few as 300 students contrast with some having more than 10,000. Administrative data processing and educational computers already existed at several of the schools.

What distinguishes this network from most earlier ones is its emphasis on faculty training. Each school was required to nominate four faculty members who would participate in the project during its two year life. An attempt was made to attract a diversity of disciplines. The participants ranged from newly appointed juniors to senior faculty members with administrative responsibilities. Their teaching interests spanned fields from astronomy to zoology. Their previous contact with computers varied from none to extensive programming work.

In order to distribute the project staff effectively, four groups were formed: physical sciences, social sciences, business, and art and humanities. These groups encompassed the following fields:

Physical Sciences

astronomy
engineering
chemistry
mathematics
physics

Social Sciences

anthropology
biology
education
geography
pharmacology
political science
psychology
sociology
zoology

Business

accounting
business administration
data processing
economics
management
marketing

Art and Humanities

art
English
music
philosophy
Spanish

These groups are not arranged according to some natural relationships which exist between the subjects, but according to the types of computer techniques most frequently needed for a particular subject. For example, the physical science group tended to express problems in analytical terms while the social science group was more likely to rely on statistical procedures. To my knowledge the Northern California Regional Computer Network is the broadest venture in computer training for college faculty ever undertaken, and the only one which has involved more than a token participation of arts and humanities instructors.

545

The foundation for this project was the month long summer session held at the Stanford Computation Center in August of 1971. The summer session presented several problems. A limited staff of two to three persons had to design a schedule which could be executed by six persons. The program had to accommodate novices and experienced computer users at the same time. Offering a series of languages, technical information, etc. was not enough. At least half of the program was to be devoted to workshops or presentations on classroom applications. Specialized topics were to be presented only to those who had use for them. The project staff, whose teaching experience was limited to short computer courses, had to review their knowledge of the subjects in the area of their responsibility and research existing computer applications. The immediate goal of the session was to impart a sufficient amount of computer knowledge to allow the participants to proceed on their own during the academic year.

The selection of computer languages was made on the basis of choosing only those which would be necessary to accomplish that goal. The result was three languages: BASIC, FORTRAN and SNOBOL. Participants learned to use IBM 2741 terminals for remote job entry, and terminals were later installed at each school for their use during the following academic year. BASIC was used as the primary vehicle to convey an understanding of programming and computers. It has the advantage of being an interactive language, which increases its effectiveness with beginners severalfold. FORTRAN was chosen to handle the larger problems and those which could not fit within the constraints of BASIC. SNOBOL was used for the arts and humanities applications. Both FORTRAN and SNOBOL were made available in highspeed versions (WATFIV and SPITBOL)[5].

This selection of languages served the project's needs very well. BASIC was taught during the first two weeks of the summer session, and FORTRAN and SNOBOL were taught at the same hour during the last half of the session. Thus it was possible to study, at most, two languages. If a participant showed difficulty with BASIC, he was encouraged to restudy and practice it rather than attempt another language.

A large number of programs in BASIC and FORTRAN were used to illustrate classroom applications of the computer and to provide models for programming. The majority of these programs were collected from the universities which have conducted networks in the past[6].

Before any actual programming could be accomplished, the operational procedures of the computer system had to be learned. In our case, this meant that a limited knowledge of the WYLBUR[7] text editor had to be acquired. This material was alternated with the BASIC language classes so that actual programming could begin as soon as possible. In the final opinion of staff and participants, the text editor added a layer of complexity which was unmanageable at that time. Many participants were confused as to which keywords belonged to BASIC and which belonged to the text editor (e.g., PRINT vs. LIST). It would have been preferable to isolate a minimal subset of the text editor commands and defer the presentation of all others until at least the second or third week.

The format of the summer session was chosen to provide variety and flexibility. The mornings were divided into three fifty minute lectures. The afternoons were termed workshops and consisted of the working groups discussing applications, problems from the morning lectures, meeting with guest professors who were using the computer in their classes and working on the computer terminals. A summary of the morning lecture hours is given in Appendix 1. They were held in a room equipped with a television camera and four overhead monitors. This allowed the output from the terminal to be displayed at the time the subject was being taught--a most effective technique.

Only fifteen IBM 2741 terminals were available in the afternoons, and this proved to be inadequate for seventy-two participants. Arrangements were made for evening use of the terminals, and eight teletype terminals were added in the second week. This change was particularly well received by those participants residing at Stanford. A staff member was usually present in the terminal room to answer questions as they arose. Even more such help (which was not feasible) would have been welcome.

Since it was possible to give only a limited amount of individual attention, adequate reference materials and study guides were deemed to be of great importance. A survey of publishers' materials and those from Stanford was made, and those listed in Appendix 2 were distributed. The proceedings of the Iowa and Dartmouth conferences as well as the ConUSE report were especially appreciated by the participants. Books which were aimed at specific disciplines were distributed only to those who showed an interest in that subject.

While most of the materials were intended for reference and study during the remainder of the year, several participants felt that too much information was being distributed for them to absorb. Repeated assurances that not everything need be read that month helped to some extent.

One of the design objectives of the summer session was to prevent technical computer subjects from dominating the entire program. Guest speakers both from Stanford and from other institutions provided a balance to the technical lectures by discussing their classroom

experiences, reporting new research activities, and giving demonstrations. Some of these presentations were made to all participants in the mornings; others were given only to the afternoon group most interested. Adequate time was allowed for extended discussion with any of the speakers. Survey classes on such subjects as computer literature (several publishers supplied free copies of their magazines) were used to give the participants some background in computing activities. Appendix 1 lists these classes and the guest lecturers.

In order to gauge the project's progress, two questionnaires were distributed, one at the end of the second week and one at the close of the summer session. Appendix 2 summarizes the responses to these questionnaires. It is satisfying to note that over 90% rated the overall effectiveness of the project good or excellent. I believe that this is partly due to responses which the project staff made to specific suggestions from the first questionnaire, e.g., allow more free time for study. On the other hand, for every suggestion made by several participants, there was at least one who made nearly the opposite request. This is not surprising if the differences in experience and interest are considered.

The project's strongest feature was the use of a comprehensive computer system with excellent software support; its weakest was the lack of specific guidance in the design of discipline oriented programs. The participants who felt least accommodated by the projects were usually from the arts and humanities group, which, incidentally, received approximately two to three times the individual help of any of the other groups. Their goals were often set too high to be accomplished in a short time. However, at least one of these individuals has become very active and interested after returning to his own school and is now working independently.

Even though the Northern California Regional Computer Network is still in its early months of operation, some of the secondary effects of the summer session are already visible. Two of the schools have held small-scale training projects for other faculty members. Over two dozen discipline oriented computer programs have been submitted to the Network's library, which is expected to double several times within the next year. Campus and local newspapers have given the project publicity. Pacific Union College is acquiring a larger timesharing computer because of increased interest by faculty in the business and social science departments. Previously, their computer had been nearly the exclusive property of the physics department. Two colleges which are not part of the Network have tied in with Stanford's computer, partly to share the results which this project will generate. It is reasonable to expect even more concrete benefits from the project in another year or two.

Many colleges and universities, even those of modest size, have the capability to do far more than they are presently doing in terms of faculty training. Some of the approaches presented in this report could be combined with the existing computer capabilities to spark new interest in the computer as an academic tool. Two or more schools could offer a joint program to attract additional support. The mere bringing together of many faculty members with a degree of common interest will start a beneficial exchange of experience and knowledge.

REFERENCES

1. Colorado State University and Washington State University also have active projects of this type.
2. NSF grant GJ-28758, "Stanford Bay Area Ed. Network for Curriculum Development."
3. The Campus Facility of the Stanford Computation Center offers batch and timesharing services with more than two dozen languages. It also provides short computer courses, consulting, contract programming and documentation.
4. The schools comprising NCRCN are: College of the Holy Names, Oakland; College of Marin, Kentfield; College of Notre Dame, Belmont; Contra Costa College, San Pablo; DeAnza College, Cupertino; Diablo Valley College, Pleasant Hill; Foothill College, Los Altos Hills; Gavilan College, Gilroy; Golden Gate College, San Francisco; Laney College, Oakland; Menlo College, Menlo Park; Mills College, Oakland; Pacific Union College, Angwin; San Jose City College; Sonoma State College, Rohnert Park; U. of the Pacific, Stockton; U. of Santa Clara, Santa Clara; West Valley Community College, Cambell.
5. WATFIV--A fast, one pass FORTRAN IV compiler for the IBM 360, developed by U. of Waterloo Computation Centre. SPITBOL (Speedy Implementation of SNOBOL4) was developed at the Information Science Center, Illinois Institute of Technology.
6. Notable contributors were: Dartmouth College, North Carolina Educational Computer Service, Illinois Institute of Technology, Beloit College and Texas A & M.
7. A summary of text editors, including WYLBUR, is found in "On-line Text Editing: A Survey," A. V. Dam and D. E. Rice Computing Surveys, V3, number 3, Sept. 1971, pp. 93-114.



APPENDIX 1

Morning Lecture Sessions

<u>Subject</u>	<u>Lecturer</u>	<u>Hours</u>
BASIC	P. Goldstein	11
WYLBUR Text Editor	J. Genis	6
FORTRAN	P. Goldstein	12
SNOBOL	C. Farlow	
Welcome, discussion of goals, etc.	R. Code	1
Network Activities in U.S.	N. Nielsen	1
Survey of Classroom uses of Computers	R. Code	1
History of the Computer	R. Carr	1
Program Documentation	K. Murata	2
Computer Literature	B. Lemm	1
Survey of Computer Languages	R. Code	1
Stanford's Program Libraries	B. Lemm	1
Job Control (JCL)	A. Kappahn	2
Computer Cost Considerations	B. Lemm	1
Getting a Program to Run	B. Lemm	1
Measurement of Effectiveness	B. Lemm	1
Networking in the Future	J. Moore	1
Plans for Next Eleven Months	R. Code	1
Reports from Group Leaders	*	2
Plan for Next Summer	R. Code	1
Closing Session	R. Code	1
Computer Graphics	J. Genis	1
Miscellaneous (short topics)		2
* <u>Group Leaders:</u> Arline Kappahn -- Physical Science Group		
Karen Murata -- Social Science Group		
Susan Patick -- Business Group		
Pat Box -- Arts & Humanities Group		

APPENDIX 1 (continued)

Guest Lectures

<u>Lecturer and Subject</u>	<u>Hours</u>
Alfred Bork, University of California at Irvine "Dialogs in Physics" with slides and demonstration	1.5
Harry Schey, Massachusetts Institute of Technology "Animated Graphics" with films	1
Herbert Peckham, Gavilan College "The Computer in a Small College"	1
Patrick Suppes, Stanford University "CAI -- The State of the Art" with film	1
Joseph Denk, North Carolina Education Computing Service "Network Curriculum Development"	1.5
Jerold Feldman, Stanford University "The Artificial Intelligence Project"	1
Erwin Parker, Stanford University "SPIRES -- Information Retrieval System" with film	1

Afternoon Guests

Physical Science Group

Alfred Bork, University of California at Irvine
Harry Schey, Massachusetts Institute of Technology
Herbert Peckham, Gavilan College
Joseph Denk, North Carolina Education Computing Service
George Homsy, Stanford University
William Weaver, Stanford University

Social Science Group

Joseph Denk, North Carolina Education Computing Service
Peter Newstead, University of San Francisco
Raymond Rees, Stanford University

Business Group

Norman Nielsen, Stanford University
William Massey, Stanford University
William Beaver, Stanford University

Arts & Humanities Group

Leland Smith, Stanford University
Laura Gould, University of California at Berkeley
Jef Raskin, University of California at San Diego

APPENDIX 2

Questionnaire No. 1 (at end of second week -- 52 replies)

1. What is your overall evaluation of the project?

Most responses were very favorable.

2. What should be changed to make improvements?

Ans.: Group participants according to prior experience. (14)*

Schedule classes for greater continuity. (6)

Allow more terminal time during the day. (5)

Give assignments for the lecture classes. (4)

Do not change anything. (4)

Slow down the pace. (3)

Discuss in greater detail the final goals. (2)

Give less attention to details. (2)

Allow more free time in the afternoons. (2)

More practice in flowcharting needed.

Allow smoking in the workshops.

Divide workshops by specific disciplines some of the time.

3. Has your enthusiasm increased, decreased, or stayed the same?

Increased: 30 Decreased: 3 Same: 13

4. Which aspects of the project have you appreciated the most?

Ans.: Helpfulness, etc. of the staff. (17)

Actual computer experience. (11)

Guest speakers. (8)

Wide scope of the project. (5)

Afternoon workshop. (5)

Study and reference materials. (4)

Effectiveness of the instruction. (4)

Actual demonstrations. (3)

Stanford's facilities. (3)

Freedom to choose among many options. (2)

*Numbers indicate the multiplicity of the responses.

APPENDIX 2 (continued)

Which the least?

Ans.: Some aspect of the afternoon workshop. (10)

Certain guest lecturers. (5)

Too fast. (4)

Too much mathematics and science. (3)

Too much detail. (2)

Classes too large. (2)

Too many disjointed activities.

Lack of staff knowledge in specific disciplines.

Breaks too long.

Coffee too strong.

5. If you are staying at Stanford, are your accommodations satisfactory?

Yes: 13 No: 2

6. Do you have sufficient terminal time?

Yes: 30 No: 20

7. Is the material too technical?

Yes: 10 No: 42

8. Are the instructors satisfactory?

Yes: 41 No: 2

9. Are more guest speakers needed?

Yes: 14 No: 34

552

APPENDIX 2 (continued)

Questionnaire No. 2 (at end of summer session -- 60 replies)

1. Please rate:	<u>Excellent</u>	<u>Good</u>	<u>Fair</u>	<u>Poor</u>
a. Performance of SCC staff.	29	27	3	0
b. Quality of the facilities.	35	22	3	1
c. Suitability of material and instruction for your requirements.	16	27	12	3
d. Overall effectiveness of the project.	17	37	0	2

2. If given the choice again, would you have begun this project?

Yes: 56 No: 4

3. During this summer I studied:

BASIC: 60 FORTRAN: 19 SNOBOL: 26

I already knew:

BASIC: 6 FORTRAN: 18 SNOBOL: 0

4. What would you like to see in next summer's program?

More terminals. (54)

More reports from participants themselves. (39)

Remedial or repeat instruction. (35)

More individual help. (35)

More work in CAI (35)

More free time. (34)

More research by group leaders into applications. (25)

More guest speakers. (17)

Use of information retrieval programs. (15)

REFERENCES

1. President's Council on Advisory Commissions. COMPUTERS in Higher Education. Washington: U. S. Government Printing Office, February 1971.
2. Stanford Computation Center. Compu Facility Users' Manual, third edition. Stanford: January 1971.
3. Stanford Computation Center. SYMBUR Reference Manual, fourth edition. Stanford: July 1971.
4. Stanford Computation Center. Introduction to Programming Concepts. Stanford: 1971.
5. Nolan, Richard L. Introduction to Computing through the BASIC Language. New York: Holt, Rinehart and Winston, Inc., 1969.
6. Stanford Computation Center. Stanford BASIC Users' Manual, third edition. Stanford: July 1971.
7. Center for Conferences and Institutes. Proceedings of a Conference on Computers in the Undergraduate Curriculum. Corvallis. University of Iowa: September, 1970.
8. Proceedings of Second Annual Conference on Computers in the Undergraduate Curricula. University Press of New England, 1971.
9. IBM Corporation. System/360 FORTRAN IV Language (GC28-6515).
10. IBM Corporation. System/360 FORTRAN IV S C M Programming Guide (GC28-6817).
11. Stanford Computation Center. FORTRAN Plus. Stanford: 1971.
12. Stanford Computation Center. System/360 BASIC Computer. Stanford: August 1970.
13. Stanford Computation Center. FORTRAN under OS/VS (timesharing FORTRAN). Stanford: August 1970.
14. Griswold, R. E., Poage, J. F., and Polonsky, I. P. The SNOBOL4 Programming Language, second edition. New York: Prentice-Hall, 1971.
15. Ashby, S. E., and Hull, C. H. Introduction to I/O Concepts and Job Control Language for the IBM Operating System 360. New York: McGraw-Hill, 1970.
16. Stanford Computation Center. Overall Plotting System. Stanford: 1971.

STUDY MATERIALS

Physical Science

1. Steinberg, Warren B. and Walker, Robert J. Calculus - A Computer Oriented Presentation CRICISAM. Tallahassee: Florida State University Press.
2. Henriksen, Melvin and Lees, Milton. Single Variable Calculus. New York: Worth Publishing Company, 1970.
3. Dartmouth Computing Project. Use of the Computer in a Course in Number Theory. Dartmouth College.
4. Dartmouth Computing Project. Using the Computer in the Teaching of Constructive Linear Algebra. Dartmouth College.
5. Peckham, Herbert. Computers, BASIC and Physics. Reading, Massachusetts: Addison-Wesley, 1970.
6. Weaver, William. BASIC Programs for Applied Mechanics. Stanford: Stanford University Press, 1970.
7. Bork, Alfred. Notions about Motions. San Francisco: W. H. Freeman Company, 1971.
8. Bork, Alfred. FORTRAN for Physics. Reading, Massachusetts: Addison-Wesley, 1966.

9. IBM Corporation. System/360 Continuous System Modeling Program -User's Manual (GH20-0367).
10. Jensen, R. and Lieberman, M. ECAP -IBM Electronic Circuit Analysis Program. New York: Prentice-Hall, 1967.

Social Science

1. Nie, N. H., Bent, D. H., and Hull, C. H. Statistical Package for the Social Sciences. New York: McGraw-Hill Company, 1970.
2. Technical Reports of the Mathematical Studies Institute for the Social Sciences. Stanford University.
3. Specifications for Social Sciences Instructional Programming Project. Beloit, Wisconsin: Beloit College, 1970.
4. Use of Computers in High Schools, teachers' manual. Brooklyn, New York: Polytechnic Institute of Brooklyn, August 1969.
5. Project IMPRESS, Interim Report. Dartmouth College: July 15, 1970.
6. IBM Corporation. Remote Access Statistical System (Share 360D 13.0.003).

Business

1. FORTRAN Applications in Business Administration. Graduate School of Business Administration, University of Michigan, 1969.
2. Couger, J. Daniel. Computers and the Schools of Business. School of Business Administration, University of Colorado, 1967.
3. Schriber, Thomas J. FORTRAN Case Studies for Business Applications. New York: John Wiley & Sons, 1969.
4. IBM Corporation. Curricula Involving Computers in Collegiate Schools of Business. Publication 320-1035-0.
5. Pillsbury, Wilbur. Computer Augmented Accounting, CompuGuide 1 & 2. Cincinnati: South-Western Publishing, 1970.
6. Alderson, W., and Shapiro, S. J. Marketing and the Computer. New York: Prentice-Hall, 1967.
7. Kanter, Jerome. The Computer and the Executive. New York: Prentice-Hall, 1967.
8. Withington, Frederick. The Use of the Computer in Business Organizations. Reading: Addison-Wesley, 1971.
9. Wohl, G., and Jauch, H. The Computer--An Accounting Tool. Homewood: Richard D. Irwin, Inc., 1965.
10. Time-Sharing Applications in Commercial Banking. Tuck School of Business Administration, Dartmouth College, June 1970.
11. Darden, B. R., and Lucas, W. H. The Decision-Making Game: An Integrated Operations Management Simulation. New York: Appleton-Century-Crofts, 1969.
12. Henshaw, R. C., and Jackson, J. R. The Executive Game. Homewood: Richard D. Irwin, Inc., 1966.
13. Graham, R. G., and Gray, C. F. Business Games Handbook. New York: American Management Association, Inc., 1969.
14. Elliott, C. O., and Wasley, R. S. Business Information Processing Systems. Homewood: Richard D. Irwin, Inc., 1971.

Arts and Humanities

1. Lincoln, Harry B., ed. The Computer and Music. Ithaca: Cornell University Press, 1966.
2. Leed, Jacob. The Computer and Literary Style. Kent: Kent State University, 1966.
3. Reichardt, Jasia, ed. Cybernetic Serendipity--the Computer and the Arts. New York: Frederic A. Praeger, Inc., 1969.
4. Reichardt, Jasia. The Computer and Art. New York: Van Nostrand, 1971.
5. Report of the Conference on Computer Technology in the Humanities. Lawrence: University of Kansas, Sept. 3 - 7, 1969.
6. IBM Corporation. IBM Symposium on Introducing the Computer into the Humanities. Poughkeepsie, N.Y., June 30-July 2, 1969.
7. IBM Corporation. Literary Data Processing Conference Proceedings. Sept. 9-11, 1964, IBM Publication 320-0906-0.
8. IBM Corporation. Introduction to Computers in the Humanities, IBM Publication GE 20-0382-0.

FRIENDLY PERSUASION: INITIATING RELUCTANT FACULTY
TO THE COMPUTER IN THE CLASSROOM

L. D. Kugler
J. M. Snider
University of Michigan
Flint, Michigan 48503
Telephone: (313) 253-7541

Introduction

Despite strenuous efforts in advertising, promotion and the news media, partisans of computer aided instruction at the collegiate level have not persuaded the majority of faculty to make use of computers in the undergraduate curriculum. In this paper, we report our evangelistic efforts at the University of Michigan-Flint, with emphasis on aspects of our campaign that we think have been especially fruitful.

The University of Michigan-Flint is a relatively small liberal arts college, with approximately 2,000 students and a staff of 90 full-time faculty members.

The UMF computing facility is located in one large room which is open twelve hours a day during the week, and five hours on Saturday. We have three terminals (two teletypes and one Datel) which have access to the University of Michigan's (Ann Arbor) Michigan Terminal System (a dual 360/67 time-sharing machine), along with an IBM 2780 card reader-printer, plus two keypunch machines. Student assistants are available at all times to assist users in operating the terminals and to run "batch jobs".

When we started our program in September, 1971, seventeen of the faculty members had had previous experience with the computer; but of these seventeen, only seven had ever used the computer in the classroom. The general attitude of the other faculty toward the computer seemed to be that it was strictly for the mathematicians and scientists, or for those who had nothing else to do but play games (football, 3-dimensional tic-tac-toe, etc.), or for those who had a lot of time and money to waste and who liked to tinker with machines.

The plan we devised to effect faculty attitude change had both short-term and long-term objectives. Our immediate goals were to overcome the fear of the computer which many faculty members had; to demonstrate its effective use in the classroom in all disciplines; and to provide "hands-on" experience to as many faculty members as possible. In the long run, we hope to see considerable expansion of faculty computer use (at least one terminal in every department), and the computer used for all routine clerical tasks, such as scoring, compiling statistics, etc., so that the teacher can be free to teach. But most of all, we would like to see the faculty incorporating effective learning models into courses in every discipline to facilitate learning for all students.

Ways and Means

Our budget was not large enough (nor was there need initially) to provide for a crew of programmers. Instead, we suggested that the college administration create the position of Computer Consultant to the Faculty. This position is held by the second author, whose responsibilities are to assist faculty in obtaining and using existing programs and documents for instructional and research purposes, to help faculty members become acquainted with programming languages appropriate to their disciplines and to arrange and demonstrate computer-assisted-instruction programs.

Our initial approach to the faculty was to distribute (about the third week of the semester) a do-it-yourself introduction and guide to the UMF Computing Facility. Included were:

1. A description of the services available to the faculty from the Faculty Consultant
2. A brief summary of the operation of our Computing Facility
3. Easy instructions on how to set up accounts (these are necessary to use our system)
4. Announcement of a Faculty Seminar on the computer as an instructional resource
5. A nontechnical introduction to an easy instructional computer language, FOIL, which was developed at the University of Michigan, Ann Arbor, and is available on our system (Michigan Terminal System, or MTS)

The packet also contained an individual computer identification code good for the entire year. Each account contained an initial credit of \$25.00, which was charged to the computing facility (a total of \$3,000) and not to the departments.

Within a week after distribution of this packet, we began our seminar program. Each seminar consisted of a short talk (about 45 minutes) followed by an on-line demonstration of at least one of the programs discussed. (We feel strongly the importance of demonstrations as an effective motivational device.) Lunch or refreshments were served to the participants, and the speakers were generally available for consultation for a full afternoon.

The first seminar was a general introduction to MTS and the concept of computer assisted instruction. This seminar was given twice to accommodate schedule conflicts on the part of the faculty. Later a summary of the actual discussion was typed and distributed to all of the faculty and staff.

The second seminar was a cross-disciplinary discussion of learning models. It included several different types of models which can be constructed on a computer (games, simulations, etc.) and a demonstration of each kind.

Four discipline-specific seminars followed, presented by invited speakers who had given papers at the Conference on Computers in the Undergraduate Curriculum at Dartmouth in June, 1971. These speakers discussed their use of the computer respectively, in an undergraduate Sociology curriculum, in an Introductory Management Statistics course, in undergraduate Economics, and for teaching strategy in scientific research (particularly in Psychology and Chemistry).

For the demonstrations, all of the guests sent materials ahead (tapes, cards, or listings of their programs) so that we could make any adaptations necessary for use in our system, or else they provided us with a phone number and an I.D. code so that we could call their system.

Publicity for the seminars was one of our biggest problems. How does one get the attention of a faculty member who is already flooded with memos, notices, and other assorted notes and papers?

We attacked this problem in several ways. We tried to be imaginative in memos sent to the faculty (for example, a large "CONFIDENTIAL" or "FOR MATURE ADULTS ONLY" stamp on the outside, or a computer printout). About one week prior to each seminar, we sent hand-written invitations to faculty members especially interested in the topic of the next seminar. One or two days before the seminar, we made phone calls or made it a point to see as many faculty and staff members as we could to extend a personal invitation to come and enjoy an afternoon with us.

In short, we used a "hard-sell" approach, but we tried not to intimidate. Our attitude was that we had a service to offer, and that learning to use the computer would not only be useful, interesting, and relatively painless, but also a very joyful and satisfying experience.

Results

Our seminars were fairly well attended. On the average, about 15 faculty members participated each time. Over the entire semester, 41 out of 90 full-time faculty members (46%) attended at least one seminar. Faculty from every department (including English and the Foreign Languages) participated in the program. Eleven faculty members (12%) actually used the computer in the classroom this semester, and 7 of these 11 were "first-time" users.

In order to give the faculty an opportunity to appraise our program, to let them evaluate the seminars, the computing facilities, express their likes and dislikes, etc., we distributed a questionnaire at the end of the Fall semester. (See Appendix I). About 80 questionnaires were passed out, 47 of which were completed and returned. The questionnaire was a type of multiple-choice in which the respondent was asked to choose all appropriate responses (sometimes more than one), and had room for comments. It took about ten minutes to complete.

The response we got was encouraging. For example, 38 faculty members have indicated that they definitely plan to use the computer in the classroom, and 6 have indicated that they "probably" will.

To the question, "What was your general impression of the introductory package that you received at the start of the semester (containing your sign on number, computing center schedule, etc.)?", 34 of the 47 replied positively. Seven actually checked, "great-everything I needed to get started".

Reaction to the FOIL manual was not as encouraging: 24 said that they didn't read it, but 22 said that they filed it for future reference - only 3 said that they threw it out. On the other hand, 25 (or 53%) of the replies stated that the respondent read or at least skinned

through the manual, and 7 (or 14%) said that they actually went to the computing center and tried it out.

Announcements of seminars by memo seemed effective: 36 of the 47 respondents said that they learned about the seminars by reading the memos. Since only 16 were reached by special written invitation, we have concluded that they may not have been worth the effort. Most of those who did not attend any seminars said that "none seemed relevant" (8 out of 18). Some of these asked for specific emphasis on the Humanities, which we are planning to include in the Spring semester. Most of the remaining replies indicated "schedule conflicts" as the cause.

We asked the respondents to evaluate each individual seminar, giving them a wide spectrum of choices. None of the seminars was rated "too long" or "of no value whatsoever". Every seminar was rated at least once as "gave me new ideas", 5 of the 6 were marked "thought maybe I'd try it out", and 2 were labeled "very exciting, expect to use this stuff right away".

We asked the respondents how the computer seminar program affected their attitude toward the computer by asking them to express their feelings before the program started and now.

The breakdown of responses is as follows:

<u>BEFORE</u>	<u>NOW</u>
3 - hated it	9 - love it
3 - mildly negative	21 - mildly positive
7 - neutral	6 - neutral
17 - mildly positive	0 - mildly negative
6 - loved it	1 - hate it

The numerical results are fairly clear. As for the nature of the individual changes, not one was less positive now than before. One checked "hated it" before and still does; but of the other 5 who either "hated it" or felt "mildly negative", 1 checked "neutral" now, and the rest are "mildly positive" or above. Most of those who were neutral (7) remained neutral (5), but 2 now feel "mildly positive". Of the 16 who checked "mildly positive", 15 still are, and 1 now "loves it".

The criticisms and recommendations offered were varied, but almost all (45 out of 47) asked for increased services. The respondents expressly asked that the introductory seminars be held again, that we increase the number of terminals available, make student programmers available, have more demonstrations, and have more guest speakers.

Summary and Future Plans

Perhaps the most significant factors that have produced such response from an initially uninvolved and reluctant faculty are the following. We tried to introduce the faculty to the computer in a joyful, relatively uncomplicated way. Our aim was not to encourage each instructor to become a programmer, nor to create computer technicians, but simply to make each one aware and more inclined to make use of CAI programs already available. To provide follow-through, we have begun a library of CAI programs, resources, and materials which we plan to continue. The FOIL manual complements this by providing a resource which will enable faculty to write their own programs in a matter of a few hours.

The actual demonstrations on the computer had more impact on the participants than anything else. Instructors who were initially somewhat inhibited or timid became enthusiastic and excited when they had the chance to sit down at a terminal and run programs themselves for the first time. This opportunity to take part in computer instruction activity should, in our opinion, be a part of any program to encourage faculty to use the computer in the classroom.

APPENDIX I

Faculty Questionnaire with Response Totals (Total questionnaires returned: 47)

1. At the beginning of this semester, how much did you know about computers and programming?
 - 12 - a. not a thing
 - 21 - b. a little, but not much
 - 10 - c. a fair amount, actually
 - 0 - d. a lot
 - 1 - e. everything there is to know

Introduction to the Computer

2. What was your general impression of the introductory package you received at the start of the semester (containing your sign on number, comp. center schedule, etc.)?

- 6 - a. what package?
- 1 - b. too technical
- 0 - c. totally inadequate
- 6 - d. somewhat inhibiting
- 4 - e. reassuring
- 19 - f. interesting
- 7 - g. great - everything I needed to get started
- 4 - h. other (specify):

3. How did you feel about the FOIL manual?

- 24 - a. didn't read it
- 6 - b. didn't need it
- 0 - c. too technical
- 0 - d. offensive - in bad taste
- 3 - e. corny
- 8 - f. good, easy introduction to computing
- 1 - g. gave me a running head start
- 0 - h. exercises were too hard
- 1 - i. exercises were irrelevant
- 4 - j. full of valuable information
- 1 - k. complete and easy to follow
- 4 - l. fun to read
- 3 - m. other (specify):

4. What did you do with the manual?

- 10 - a. skimmed it - intend to look at it more
- 8 - b. read it
- 7 - c. went to the Computing Center and tried it out
- 3 - d. threw it out
- 22 - e. filed it for future reference
- 0 - f. other (specify):

Seminars

5. How did you learn about the seminars?

- 36 - a. read the memos
- 7 - b. read the signs posted
- 16 - c. special written invitation
- 9 - d. read the "Daily News"
- 6 - e. a colleague informed me
- 0 - f. other (specify):

If you attended one or more of the computer seminars this semester, skip to Question 7.

6. If you did not attend any seminars, why not?

- 6 - a. schedule conflicts
- 8 - b. none seemed relevant
- 0 - c. wasn't notified in time
- 1 - d. whole thing seemed a waste of time
- 3 - e. other (specify):

Skip to Question 8

7. How would you evaluate the seminars you attended?

- 16 - a. attended part, but not all
- 15 - b. attended all
- 1 - c. too technical
- 0 - d. too long
- 4 - e. boring
- 0 - f. of no value whatsoever
- 19 - g. somewhat interesting

- 25 - a. gave me new ideas
- 16 - a. thought maybe I'd try it out
- 6 - b. very exciting, expect to use this stuff right away

Using the Computer in the Classroom

If you have used the computer in class, skip to Question 9.

8. If you have never used the computer in your class, it is because you:

- 16 - a. didn't have time to work it in
- 14 - b. didn't know how
- 1 - c. wanted to, but couldn't get help
- 7 - d. couldn't find appropriate programs
- 5 - e. could never see any potential in it
- 3 - f. just never wanted to
- 1 - g. feel computers are dehumanizing
- 1 - h. other (specify):

skip to question 13.

9. What kinds of computer exercises did your students do?

- 2 - a. multiple-choice
- 4 - b. games
- 4 - c. writing of programs for computational problems
- 4 - d. doing computational problems using programs which you provided
- 0 - e. discussion
- 1 - f. simulation of experiments
- 1 - g. analysis of data
- 3 - h. other (specify):

10. Where did you get the programs you used?

- 5 - a. wrote your own (with no help)
- 4 - b. wrote your own (with help)
- 3 - c. had someone else write them for you
- 2 - d. used "canned" programs
- 0 - e. other (specify):

11. On the average, how much class time did you spend per week preparing and carrying out computer-assisted instruction?

- 4 - a. less than 10 min/wk
- 3 - b. 10-30
- 0 - c. 30-45
- 0 - d. 45-60
- 0 - e. 1-2 hours/wk
- 1 - f. 2 hours/wk or more

12. Were the computer exercises you assigned:

- 6 - a. optional
- 5 - b. required

General Recommendations

13. How did the computer seminar program affect your attitude toward the computer?

<u>BEFORE</u>	<u>NOW</u>
3 - a. hated it	9 - a'. love it
3 - b. mildly negative	21 - b'. mildly positive
7 - c. neutral	6 - c'. neutral
17 - d. mildly positive	0 - d'. mildly negative
6 - e. loved it	1 - e'. hate it

14. What criticisms or recommendations do you have about UMF's instructional computing facilities?

- 8 - a. get more speakers in (specify interest area, if possible):
- 15 - b. hold the introductory sessions again
- 13 - c. get more terminals
- 12 - d. make student programmers available
- 1 - e. extend the hours that the computer room is open (specify when):
- 6 - f. provide a private faculty-staff terminal
- 10 - g. have more demonstrations
- 5 - h. get more consultants
- 2 - i. eliminate this computer nonsense altogether
- 15 - j. other comments:

APPENDIX II

One of the questions on the faculty questionnaire deals with the extent to which the computer has already been used in the classroom at UMF.

Most of the faculty who said that they had never used the computer in the classroom "didn't have time to work it in" or "didn't know how". Others said that they "couldn't find appropriate programs." Five checked that they "just never wanted to." Only one checked "feel computers are dehumanizing"!

A questionnaire was circulated to students in courses where the computer was used. (This included courses in Physics, Math, Chemistry, Accounting, Finance, and History). Although the results are not directly relevant to the subject of this paper, they are of interest as an indicator of student attitude response to the use of the computer in the classroom. 108 questionnaires were distributed and 141 were returned. The students were asked to check all appropriate responses. A copy of the questionnaire with the response totals is presented below.

STUDENT QUESTIONNAIRE

1. How much did you know about computers and programming before you took this class?
 - 55 - a. not a thing
 - 54 - b. a little, but not much
 - 19 - c. a fair amount, actually
 - 5 - d. a lot
 - 1 - e. everything there is to know
2. How do you regard your introduction to the computer?
 - 36 - a. totally inadequate
 - 12 - b. too technical - very hard to understand
 - 62 - c. good, as far as it went
 - 23 - d. fairly sound
 - 5 - e. complete and easy to follow
 - 0 - f. absolutely brilliant
3. On the average, how many minutes/week did you spend using the computer?
 - 36 - a. 10 minutes/week or less
 - 22 - b. 10-20
 - 14 - c. 30-45
 - 8 - d. 45-60
 - 3 - e. 1-2 hours
 - 8 - f. 2 hours or more
4. What type of exercises did you do on the computer?
 - 42 - a. multiple-choice
 - 35 - b. games
 - 27 - c. writing of programs for computational problems
 - 56 - d. doing computational problems using programs provided by the instructor
 - 5 - e. discussion
 - 28 - f. simulations of experiments
 - 32 - g. analysis of data
 - 6 - h. other (specify):
5. What did you like best about the computer exercises?
 - 59 - a. they eliminated tedious computation
 - 12 - b. they provided the opportunity for the user to perform his own analysis

- in his own way
- 45 - c. they gave instant feedback
- 23 - d. they could be repeated as often as the user wished
- 37 - e. they could be done at the user's convenience
- 8 - f. They made it possible to do things which would be impossible to do in real life
- 15 - g. they clarified the subject matter of the course
- 27 - h. they were fun to run
- 8 - i. other (specify):

6. The worst things about using the computer in this course were:

- 31 - a. the programs didn't run
- 15 - b. the programs were full of errors
- 16 - c. the assignments were completely irrelevant to the subject matter
- 6 - d. the assignments were so complex that I couldn't follow them
- 4 - e. the assignments were too long
- 8 - f. the problems bored me
- 27 - g. I could never get a free terminal
- 8 - h. the keypunches were always busy
- 11 - i. I couldn't get help when I needed it
- 15 - j. I just don't like machines
- 19 - k. other (specify):

7. If you were teaching this course, what would you do to improve it?

- 11 - a. eliminate this computer nonsense, altogether
- 71 - b. make sure my students were better prepared to use the computer
- 30 - c. use the computer more
- 31 - d. use the computer in different ways
- 32 - e. make computer exercises optional, but not required
- 21 - f. make at least some computer exercises required
- 1 - g. there's no way to improve it - sheer genius
- 7 - h. other (specify):

8. How did this course affect your attitude toward the computer?

<u>BEFORE</u>	<u>NOW</u>
4 - a. hated it	15 - a'. love it
19 - b. mildly negative	68 - b'. mildly positive
76 - c. neutral	28 - c'. neutral
33 - d. mildly positive	14 - d'. mildly negative
9 - e. loved it	10 - e'. hate it

A COMPUTERIZED PHYSICS LABORATORY

W. Thomas Bais
Macon Junior College
Macon, Georgia 31206
Telephone: (912) 745-8551

Introduction

Thanks to a summer made available for studying computer application to physics(1), the author has developed a computer-based instructional module for use in a non-calculus, non-major physics laboratory.

The purpose of this module is three-fold: (1) to reduce the traditional tutorial inefficiency of the laboratory experience caused by collaboration (e.g. one lab partner doing all the work); (2) to provide an environment more closely resembling real-life experiments; and (3) to eliminate end-of-lab cutoff by providing a means of continuing experiments from week to week.

The module requires moderate amounts of on-line access to the computer during labs. Equipment consists of one small air track with accessories (including spark device) per five or so students.

The laboratory has been used one quarter at Macon Junior College in the Georgia core curriculum non-calculus two-quarter course.

The Laboratory

The experiments are basically a series of inclined air track observations using various glider configurations. The falling gliders are given air friction and spring action in the latter part of the lab. The agenda for the student follows this order:

1. Pre-lab assignment: this involves viewing a film loop for pucks on an inclined air table and setting up and calculating values for a table containing velocity and acceleration for each timed position of the glide. The student is then introduced to the computer terminal, receiving random quiz questions from a stored file.
2. Approval of part (1) allows the first laboratory to begin using the inclined air track and specified slope (different for each student). A choice of hand calculations or canned computer program is offered for finding computer acceleration during each time segment (90% of students have chosen the computer immediately).
3. Fairly constant accelerations for part (2) allow proceeding to the computer calculation of this phenomenon. A sample program is provided by mimeograph. The student chooses appropriate initial values for x , t , v , g , etc. He multiplies g by a suitably chosen ratio related to his incline. Satisfactory comparison of the theoretical result to the air track data permits moving on.
4. An "ear" is attached to the glider. This is a piece of cardboard with a specified area (approximately 300 square centimeters). After data-taking, a term " $-cv$ " is added to the acceleration of the previous computer program. The constant " c " is adjusted until the data match.
5. Instructions suggest removing the " cv " addition and trying " $-kx$ " instead. Smaller " dt " is suggested for those who obtain nonsense.
6. Observed oscillation sends the student to the lab to try a glider attached at the air track ends by elastic sewing thread (an excellent "weak" spring). The glider falls from the "comfortable" position (with level air track). The computer program's " k " is adjusted to fit the result.
7. As an option, the student can try gravity + friction + springs. Both " c " and " k " are adjusted.

The computer usage in the procedure is quite unsophisticated. The idea is a much-used one (e.g. ref. 2) - using the equations

$$v = v + a(dt)$$

and

$$x = x + v(dt)$$

to calculate position versus time by incrementing time. The student is not given the fundamentals necessary to write such a program on his own, rather he copies a simple version and learns how it works by making appropriate changes during his progress through the lab-segments.

Three canned programs are also used; one which removes the drudgery of calculating accelerations from the air track data, and one which smooths when necessary by averaging adjacent data points. Another program administers the quiz of step 1.

Instructions consisting of one or two pages were produced for each section of the experiment. The first set included simple information for signing on and off the available computers (at that time, an IBM 360/65 and a CDC 6400 at the University of Georgia).

Student Trial

There were twenty-five students in the trial class. All reports were oral. A copy of the next segment of the lab was handed out upon satisfactory completion of the part at hand. There was little collaboration. Some students helped others with the eccentricities of the computer. Some who learned the hard way about things such as precision in elevating the air tracks passed on the experience.

The air friction segment was more interesting than expected because of the abundance of air currents coming from the track and also from a bleeder hole in the supply hose. Appropriate adjustments were required of the student to obtain consistent results.

The laboratory was well received by the students. The grade distribution was somewhat spread out compared to the usual lab. The instructor had a much firmer base for judgment than the traditional lab. Two students completed only one section, while several asked for optional experiments along the way and at the end.

The lab is quite "automated", requiring little explanation from the instructor. This allows such needed freedom to ask probing questions.

REFERENCES

1. National Science Foundation Summer Institute
2. Introductory Computer-Based Mechanics,
College Commission on Physics

MEASUREMENT OF AN AUTOMOBILE'S FUEL CONSUMPTION, ROAD HORSEPOWER,
MAXIMUM SPEED AND MAXIMUM ACCELERATION

(A Computer Assisted Laboratory Exercise Suitable for Use in
Mechanics Courses Taught at the Junior College Level)

Don Leslie Lewis
Box 162
Pettus, Texas 78146
Telephone: (512) 358-3130

Physics teachers expend much effort in devising laboratory exercises for elementary mechanics that are illustrative of the experimental foundations for classical mechanics. Physics students exert themselves to make the measurements obtained fit student notions of the mechanics principle purportedly being illustrated. Principles are thus illustrated, even decorated, and mayhaps remembered, but not used. Measurements are forgotten. A physics teacher is therefore motivated to search for realistic laboratory exercises that allow measurements the student can be expected to value.

This paper describes a procedure whereby with computer assistance students may predict commonly accepted measures of an automobile's performance using kinematic data students obtain at legal highway speeds. The exercise described in this paper has been used in physics classes at the junior college level. Computer programs to which the students had access were written by the instructor in a basic Fortran language. Students participating in the exercise were not uniformly familiar with Fortran. Unfamiliarity with the programming language is not believed to have compromised fulfillment of the single instructional objective of the exercise--to involve students in the use of selected principles from mechanics.

An estimate of the mechanical power output of an automobile engine at a particular highway speed was made from the instantaneous rate at which the kinetic energy of the vehicle diminished if the engine power input was discontinued. Gasoline consumption at a specified speed was inferred from an assumed thermodynamic efficiency for the engine, from the heat of combustion of the fuel, from the mass density of the fuel and from the mechanical power being developed by the engine at the speed of interest. The maximum attainable speed was predicted by determining a speed at which energy was dissipated at a rate equal to the automobile's maximum brake horsepower. Power limited acceleration characteristics were estimated from the difference between the rated engine horsepower and the power required to sustain a particular speed. In order to accommodate student expressed desires for a printout correlating road speed and horsepower with engine rpm, values for the rear wheel diameter and rear axle ratio were needed. Provision was made for calculation of an rpm-limited road speed.

Speed versus elapsed time data were acquired by students from speedometer readings and a stopwatch as the automobile of interest decelerated from 70 miles per hour with the ignition system off and the transmission in highway cruising condition. Measurements reported by a student are given in Table I.

Students selected values for parameters relating to a specific automobile from many sources. The numerical values for weight, maximum horsepower, and rear axle ratio listed in Table II were supplied by the student whose data is listed in Table I. This student probably furnished other participants with information included in published source[1] of which he made use. The rear wheel diameter reported in Table II was a measured value.

There are published lists of the chemical and physical properties of typical gasoline fuels[2]; but in order to give students practice in dimensional analysis and to emphasize the dimensional equivalence of heat and mechanical work, 2, 2, 4-trimethylpentane (an octane rating standard for motor fuels) was assumed to have properties similar to commercially available gasoline. Students then obtained the needed chemical and physical properties of the fuel from a readily available reference[3]. The same reference also provided students with units conversion factors. It was the responsibility of each student to supply the numerical values which are listed in Table II.

The thermodynamic efficiency of an internal combustion engine is cleverly deduced in at least one elementary physics text that has been widely used[4]. Sears and Zemansky conclude that for a 10:1 compression ratio a thermodynamic efficiency greater than 0.60 is not to be expected. Consideration of a lower compression ratio and the difference between the heat of combustion of the fuel to CO_2 and $\text{H}_2\text{O}(\text{liq})$ as products and the heat of combustion of the fuel to CO and $\text{H}_2\text{O}(\text{g})$ (the more likely products at the elevated reaction temperature) provide grounds for using a thermodynamic efficiency of 0.30. Students were aware of the arguments presented by Sears and Zemansky but were at liberty to select a value for the thermodynamic efficiency. Seeking a gasoline consumption consistent with their prejudices, roughly one-fourth of the participants

used the programs more than once with different values for the thermodynamic efficiency and heat of combustion. The writer was gratified by their enthusiasm.

The output format for the three programs referred to in Fig. 1 are illustrated in Tables III, IV, and V. The Fortran instructions used to generate Tables III, IV and V are not included in this paper. The underlying assumption and the arithmetic performed are believed most easily revealed using the symbols listed in Table II in the context of conventional mathematical notation.

Equation 1 represents a choice for the functional form of an empirical equation describing the measurements listed in Table I.

$$S = C e^{-bt} \quad (1)$$

The functional form of Eq. 1 could be anticipated if the combined forces retarding motion were directly proportional to the linear speed. Such situations probably do not persist over any wide range of speeds. Conclusions based on Eq. 1 are thus additionally suspect at speeds not listed in Table I.

The rate of decrease in kinetic energy of the car with zero power input should represent the power required to sustain motion at the given speed. Thus, giving due attention to units,

$$T = \frac{w u_1^2 S^2}{2g} \quad (2)$$

while

$$P = -u_2 \frac{dT}{dt} \quad (3)$$

so that upon using Eq. 1

$$P = \frac{u_1^2 u_2 b w S^2}{g} \quad (4)$$

The engine rpm is related (with no slip) to the linear speed by

$$r = \frac{u_1 u_3 S}{\pi d} \quad (5)$$

so that if an upper limit for engine rpm is specified, a corresponding rpm-limited speed can be determined:

$$S_{rpm} = \frac{\pi d r_{max}}{u_1 u_3} \quad (6)$$

An expression for a maximum speed limited by the available power follows from Eq. 4:

$$S_{max} = \sqrt{\frac{g P_{max}}{u_1^2 u_2 b w}} \quad (7)$$

By identifying the rate at which fuel is used at a specified efficiency with the rate at which mechanical work is accomplished against frictional forces, one can write

$$P = \frac{u_4 u_5 H \eta S}{M G} \quad (8)$$

Equation 4 can be used with Eq. 8 to formulate a convenient relationship between gasoline consumption and speed:

$$G = \frac{u_4 u_5 g H \eta S}{u_1^2 u_2 M b w S} \quad (9)$$

If energy is supplied at a given speed at a rate greater than is necessary to maintain the given speed, the automobile should accelerate. The excess power supplied, P_{XS} simply represents the rate at which kinetic energy is increased during acceleration:

$$P_{XS} = \frac{u_1 u_2 w A S}{g} \quad (10)$$

Upon assuming that the maximum power available for acceleration purposes is given by

$$P_{XS} = P_{MAX} - P \quad (11)$$

and as a consequence of relationships asserted in Eq. 4 and in Eq. 10, the maximum attainable acceleration can be expressed as a function of speed:

$$A = \frac{g P_{MAX} - u_1^2 u_2 b w S^2}{u_1 u_2 w S} \quad (12)$$

The relationship between maximum acceleration and speed given in Eq. 12 can be used to calculate a minimum time necessary for accomplishing a stipulated change in speed. Requiring that

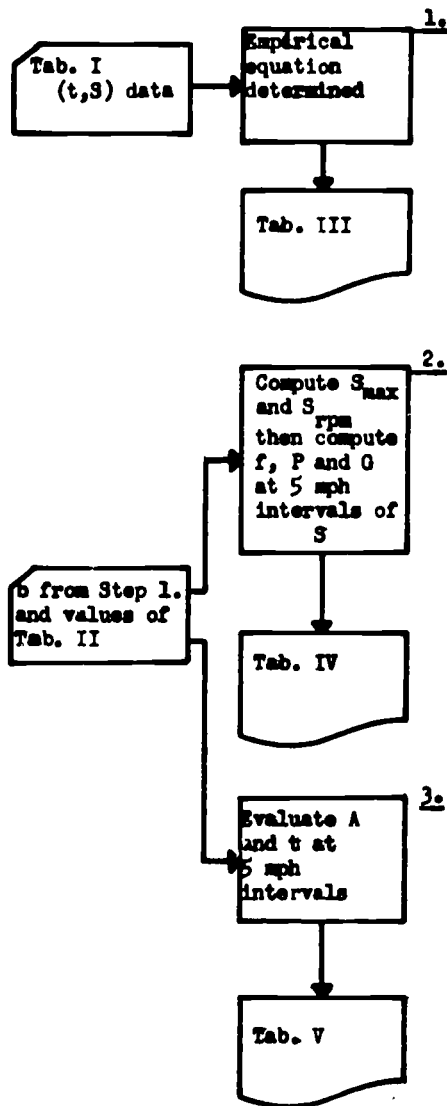
$$u_1 dS = A dt \quad (13)$$

and employing Eq. 12, one can show that

$$\Delta t = \int_{S_1}^{S_2} \frac{u_1^2 u_2 w S}{g P_{MAX} - u_1^2 u_2 b w S^2} dS \quad (14)$$

Implicit relationships between the several quantities appearing in the preceding equations have been ignored and it is convenient to consider Eq. 14 as an exercise in elementary integration techniques. Thus

$$\Delta t = \frac{1}{2 b} \log_e \left\{ \frac{g P_{MAX} - u_1^2 u_2 b w S_1^2}{g P_{MAX} - u_1^2 u_2 b w S_2^2} \right\} \quad (15)$$



Step 1. A least squares curve fitting routine appropriate to the functional form stated in Eq. 1

Step 2. Use is made of Eq. 4, Eq. 5, Eq. 6, Eq. 7, and Eq. 9.

Step 3. Use is made of Eq. 12 and Eq. 15.

FIGURE 1. Modified flow chart indicating sequence of programs used.

569

time (sec)	speed (miles per hour)
0.00	70.0
2.43	65.0
5.76	60.0
9.29	55.0
13.22	50.0
17.72	45.0
23.02	40.0

TABLE 1. Reported kinematics of a 1971 Gremlin.

Symbol	Description	Units or values
S	speed	miles per hour
t	time	seconds
T	kinetic energy	ft-lbs
P	power	horsepower
w	weight	lbs
P _{max}	rated brake horsepower	135 horsepower
S _{max}	maximum speed using P _{max}	miles per hour
S _{rpm}	rpm-limited speed	miles per hour
r	rear axle ratio	3.08
f	engine rpm	min ⁻¹
f _{max}	upper limit for f	min ⁻¹
e	thermodynamic efficiency	0.20
H	heat of combustion	1200 Kcal/mole
ρ	mass density at 20°C	0.6918 grams/ml
M	(gram) molecular weight	114.0 gram per mole
u ₁	units conversion factor	1.467 ft-sec ⁻¹ /mile-hr ⁻¹
u ₂	units conversion factor	.001802 horsepower/ft-lb-sec ⁻¹
u ₃	units conversion factor	60 sec/min
u ₄	units conversion factor	3785 ml/gallon
u ₅	units conversion factor	1.561 x 10 ⁻³ horsepower-hours/Kcal
b	an empirical constant	sec ⁻¹
d	rear wheel diameter	2.36 ft
Δt	a time interval	sec

TABLE 2. Variables and constants of Figure 1. Values listed are those used by the student whose data are displayed in Table 1.

```

1971 GREMLIN SPEEC -TYPE DATA
TYPE 7 MEASUREMENTS
C.CCCC 70.CCCC
5.7600 65.CCCC
2.4300 65.CCCC
9.2900 55.CCCC
13.2200 50.CCCC
17.7200 45.CCCC
23.0200 40.CCCC
Y# 65.1803*EXPONENTIAL -C.C24153*X WITH ERRCH 1.8525
    
```

TABLE 3. Output of step 1 of Figure 1.

 1971 CHEVILIN , 2800 LB WT, REAR AXEL RATIO 3.08 , MAX HP 135 , RPM 4000
 ISCCOT.AE- 114 GMCL WT, 12CC KCAL/MCLE, 0.6518 GRAMS/ML
 UNITS CONVERSIONS- 3785 ML/GAL, 550 FT-LBS PER HORSEPOWER-SEC, 3600 SEC/HOUR
 8.58 HORSEPOWER-FCRS/GAL

AT 20.0 % EFFICIENCY

83.6	HORSEPOWER	AT	4000	RPM	TO	MAINTAIN	100.5	MPH	WITH	10.3	MILES/GAL
135.0	HORSEPOWER	AT	5084	RPM	TO	MAINTAIN	127.7	MPH	WITH	8.1	MILES/GAL
100.2	HORSEPOWER	AT	4380	RPM	TO	MAINTAIN	110.0	MPH	WITH	9.4	MILES/GAL
51.3	HORSEPOWER	AT	4181	RPM	TO	MAINTAIN	105.0	MPH	WITH	9.9	MILES/GAL
82.8	HORSEPOWER	AT	3581	RPM	TO	MAINTAIN	100.0	MPH	WITH	10.4	MILES/GAL
74.7	HORSEPOWER	AT	3782	RPM	TO	MAINTAIN	95.0	MPH	WITH	10.9	MILES/GAL
67.1	HORSEPOWER	AT	3583	RPM	TO	MAINTAIN	90.0	MPH	WITH	11.5	MILES/GAL
55.8	HORSEPOWER	AT	3384	RPM	TO	MAINTAIN	85.0	MPH	WITH	12.2	MILES/GAL
53.0	HORSEPOWER	AT	3185	RPM	TO	MAINTAIN	80.0	MPH	WITH	12.9	MILES/GAL
46.6	HORSEPOWER	AT	2986	RPM	TO	MAINTAIN	75.0	MPH	WITH	13.8	MILES/GAL
40.6	HORSEPOWER	AT	2787	RPM	TO	MAINTAIN	70.0	MPH	WITH	14.8	MILES/GAL
35.0	HORSEPOWER	AT	2588	RPM	TO	MAINTAIN	65.0	MPH	WITH	15.9	MILES/GAL
29.8	HORSEPOWER	AT	2389	RPM	TO	MAINTAIN	60.0	MPH	WITH	17.3	MILES/GAL
25.0	HORSEPOWER	AT	2190	RPM	TO	MAINTAIN	55.0	MPH	WITH	18.8	MILES/GAL
20.7	HORSEPOWER	AT	1991	RPM	TO	MAINTAIN	50.0	MPH	WITH	20.7	MILES/GAL
16.6	HORSEPOWER	AT	1792	RPM	TO	MAINTAIN	45.0	MPH	WITH	23.0	MILES/GAL

TABLE 4. Output of step 2 of Figure 1.

 1971 CHEVILIN , 2800 LB WT, REAR AXEL RATIO 3.08 , MAX HP 135 , RPM 4000
 AT 75.0 MPH MAX ACC IS 5.05 FT/SEC/SEC

CR FROM 75.0 MPH TO 80.0 MPH IN 1.6 SEC

AT 70.0 MPH MAX ACC IS 5.78 FT/SEC/SEC

CR FROM 70.0 MPH TO 75.0 MPH IN 1.4 SEC

AT 65.0 MPH MAX ACC IS 6.55 FT/SEC/SEC

CR FROM 65.0 MPH TO 70.0 MPH IN 1.2 SEC

AT 60.0 MPH MAX ACC IS 7.51 FT/SEC/SEC

CR FROM 60.0 MPH TO 65.0 MPH IN 1.0 SEC

AT 55.0 MPH MAX ACC IS 8.57 FT/SEC/SEC

CR FROM 55.0 MPH TO 60.0 MPH IN 0.9 SEC

AT 50.0 MPH MAX ACC IS 9.80 FT/SEC/SEC

CR FROM 50.0 MPH TO 55.0 MPH IN 0.8 SEC

AT 45.0 MPH MAX ACC IS 11.26 FT/SEC/SEC

CR FROM 45.0 MPH TO 50.0 MPH IN 0.7 SEC

 MAXIMUM ACCELERATION COMPLETED FROM MAXIMUM HORSEPOWER AVAILABLE

TABLE 5. Output of step 3 of Figure 1.

REFERENCES

1. "Specifications for 1971 Passenger Cars," Motor, (February 1971), pp. 70-72.
2. Virgil B. Guthrie (Editor), Petroleum Products Handbook (New York, McGraw-Hill Book Company, Inc., 1969), p. 33.
3. Robert C. Weast (Editor), Handbook of Chemistry and Physics 51st Edition (Cleveland, Ohio, The Chemical Rubber Company, 1970-71)
4. Francis Weston Sears and Mark W. Zemansky, University Physics (Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1964), p. 429.

INFINITE SEQUENCES AND SERIES VIA THE COMPUTER

John P. Tucciarone
St. John's University
Jamaica, New York 11432
Telephone: (212) 969-8000, ext. 286

Introduction

This paper describes a program for using the computer as an aid in the teaching of infinite sequences and series to an Intermediate Calculus class in college. The course was taken by 24 first-term Sophomore mathematics majors at St. John's University, and the subject matter was designed to prepare the students for a rigorous Junior year course in Advanced Calculus.

All the students had been taught the essentials of the BASIC programming language before entering the course, and they had experienced the running of self-written programs such as the Trapezoidal Method for approximate integration, Newton's Method for approximating roots, etc. The students had access to a remote teletype terminal which accepted programs written in BASIC or FORTRAN. (Although FORTRAN had not been taught formally, three students in the class were familiar with it and wrote their programs in that language.)

Classes were devoted mainly to the teaching of abstract concepts and the computer was used as support in two ways: (1) a program would be assigned to clarify by illustration a particular property of sequences or series; (2) a program would be assigned to point up a property not yet discussed in class. In the latter case, the "discovery" by the student of a property, would then lead quite naturally to questions of the type: Does the property hold in all cases? If it does, how might a rigorous proof be constructed? This latter technique was used as a motivating force to help with the introduction of new concepts. It also acted as a means of stimulation of the students' curiosity.

A Sampling of Topics

The format of the course, viz., parallel class lectures and program assignments, worked well. As a consequence, a large number of properties were investigated via the computer. Since it is the author's intention to give the reader some idea of how to integrate the computer in the study of infinite series, only a sample of the properties will be listed here. Additional topics are suggested at the end of the paper.

1. **Convergence of Sequences:** The formal definition of convergence of a sequence $\{S_n\}$ was presented in class. That is, $\epsilon > 0, \exists N > 0$, such that for $n > N, |S_n - S| < \epsilon$. Methods were then introduced for determining whether or not a given sequence was convergent. Most of these involved calculation of the limit of S_n . Students were asked to write several programs to determine whether or not a sequence was convergent by listing enough terms to detect a pattern. Elementary sequences were suggested at the start, yet sequences for which the answer was not immediately obvious.

Example A: Does the sequence $\sqrt{2}, \sqrt{2 + \sqrt{2}}, \sqrt{2 + \sqrt{2 + \sqrt{2}}}, \dots$ converge?

Comment: Most remote teletypes which return 6 digits, will show that by the 10th term, the values are very close to 2, and continue at 2 thereafter, indicating that the sequence converges to 2.

Example B: Does the sequence $\{(1 + 1/n)^{n+1}\}$ converge?

Comment: In this program, slightly more than 1000 terms were required before the terms approached 2.718, the approximate value of the irrational number e .

2. **The Cauchy Criterion:** Once the definition of limit of a sequence was understood, the students were asked to write a program which would choose at random, 2 terms of a convergent sequence beyond a certain N th term, and test the absolute value of their difference. They were asked to select an increasing sequence of N 's. (Limitations on the system being used, and the sequence being tested, would necessitate restricting the total number of terms to be considered.)

Example C: For the sequence $\{1/n\}$, print a list of the absolute value of the difference of 2 terms chosen at random beyond the 5th, 10th, 15th, . . . terms.

Comment: Students used the random number function, RND, to select 2 terms beyond a given N th. Thus, to select 2 integer values from the 10th to the 100th, they wrote: LET K = INT[90*RND(1) + 6], the argument of RND being arbitrary. The resulting

sequence of differences indicated that as $N \rightarrow \infty$, $|S_m - S_n| \rightarrow 0$, for $m, n > N$. The students were then asked to prove that if a sequence converges to a limit, then it must satisfy the Cauchy property.

- J. Special Sequences: Consideration was given in the lectures to specific sequences which would be important throughout the term's work in the course. One of several treated was the Fibonacci Sequence. It was defined as a sequence $\{S_n\}$, where $S_1 = S_2 = 0$; $S_k = S_{k-1} + S_{k-2}$, for $k > 2$. Students were then asked to observe certain properties of the sequence, propose a theorem for each, and prove the theorem.

Example D: For the Fibonacci Sequence $\{S_n\}$, compute the sequence of values $\{S_n/S_{n+1}\}$; for selected n , compute $(a^n - b^n)/\sqrt{5}$, where $a = (1 + \sqrt{5})/2$, and $b = -1/a$. In each case propose a theorem and prove it.

Comment: In part 1, students were made to observe that if $\{S_n\}$ is the Fibonacci Sequence, then S_n may be obtained by direct substitution in the formula. In part 2, students were made to observe that the sequence $\{S_n/S_{n+1}\}$ is convergent, and its limit is approximately 0.618034.

4. Convergent Series: The definition was introduced in class in the usual manner, i.e., $\sum_{n=1}^{\infty} a_n$ is said to be convergent if the sequence $\{S_n\}$ is convergent, where $S_n = \sum_{k=1}^n a_k$. Since the definition involves the notion of convergent sequence, more exercises like A and B were introduced. When the topic of power series was introduced, the students were taught how to construct a Taylor Series expansion for a function, and how to determine its interval of convergence. They were asked to write a program that would evaluate the Taylor Series expansion of a function at a set of points in the interval of convergence.

Example E: Evaluate the Taylor Series for $\tanh^{-1}x$ at selected points in its interval of convergence.

Comment: The series for $\tanh^{-1}x$ is not a difficult one. It has the form: $x/1 + x^3/3 + x^5/5 + \dots$, and the interval of convergence is $(-1, 1)$. The students evaluated the series for $x = .1, .2, \dots, .9$ to within a given accuracy, and were asked to print not only the value obtained, but also the number of terms of the series required for the given level of accuracy. From this, they were able to see how the convergence grows weaker as one moves from the center of the interval of convergence, to the endpoints.

5. Rearrangements of Series: When the distinction between an Absolutely Convergent series and a Conditionally Convergent series had been made, and examples of each had been treated in class, the question of whether a rearrangement of the terms of a series had any effect on its convergence was considered. A given series $\sum a_n$ was divided into 2 series, the positive a_n and the negative a_n , in the following manner: $p_n = a_n$ if $a_n > 0$; $p_n = 0$ if $a_n \leq 0$. Also, $q_n = a_n$ if $a_n < 0$; $q_n = 0$ if $a_n \geq 0$. It was then proved that if $\sum a_n$ is Absolutely Convergent, then both $\sum p_n$ and $\sum q_n$ are convergent. The students were also shown that if $\sum a_n$ is Conditionally Convergent, then $\sum p_n$ and $\sum q_n$ must both diverge, the first to $+\infty$, the second to $-\infty$. The following program was then assigned.

Example F: Write a program which, by suitably rearranging the terms of a Conditionally Convergent series, will make it converge to any preassigned number.

Comment: It was suggested that as an example, the series $\sum_{n=1}^{\infty} (-1)^n 1/n$ to be used, and that it be made to converge to values such as .85, .95, 1.2, etc. The purpose of the program was to help the students realize that the theorem embodied in this exercise is valid. The values .85, etc. were suggested to avoid unnecessarily lengthy printouts. Even for the small values suggested, fairly long printouts were required (100-150 terms).

6. Summability of Series: A distinction was pointed out in the lectures between two kinds of divergent series, one like $\sum 1/n$, for which the sequence of partial sums goes to ∞ as n goes to infinity, and one like $\sum (-1)^n$ for which the sequence of partial sums oscillates from 1 to 0 as n goes to infinity. The students were shown why no limit may exist in the second case, but they were asked to consider the possibility of choosing a kind of average value for the sequence of partial sums. More precisely, given the series $\sum a_n$ and its sequence of partial sums $\{S_n\}$, another sequence $\{S'_n\}$ was defined, where $S'_n = \frac{1}{n} \sum_{i=1}^n S_i$.

Example G: Write a program to list, side by side, the terms of the sequence $\{S_n\}$ and of the sequence $\{S'_n\}$, for the series $\sum 1/2^n$.

Comment: This exercise made the students aware of the fact that as $n \rightarrow \infty$ in this case, S_n will approach 0, although not as rapidly. They were asked to try the same program on other divergent series of their choice, including both Absolutely Convergent and Conditionally Convergent series. They were asked to propose a theorem which would explain the phenomenon, and prove it. All were able to state the theorem: If $\sum a_n$ is convergent, and $\lim_{n \rightarrow \infty} S_n = S$, then $\lim_{n \rightarrow \infty} (S_1 + S_2 + \dots + S_n)/n = S$. Half were able to supply their own proofs.

When the validity of this theorem had been demonstrated to the students, another question was raised. Is the converse of the theorem also valid? Students were asked to try the same program for the series (a) $\sum 2^n$, (b) $\sum (-1)^n$, (c) $\sum (-1)^{n-1}n$. Their printouts indicated the following: in (a) both $\{S_n\}$ and $\{S'_n\}$ approached infinity; in (b) $\{S_n\}$ oscillated between 1 and 0, while $\{S'_n\}$ approached 0; and in (c) both $\{S_n\}$ and $\{S'_n\}$ oscillated. The students were able to conclude that the average method described above worked for some divergent series, but not for all. This was enough to show that the converse of the theorem was not valid. The students were then asked to consider series (c) above and a second average technique, viz., the sequence $\{S''_n\}$, where $S''_n = (n^2 + (n-1)S_2 + \dots + 2S_{n-1} + S_n)/(1+2+\dots+n)$.

Example 4: Write a program to list, side by side, the terms of the sequences $\{S_n\}$, $\{S'_n\}$ and $\{S''_n\}$, for the series in (c).

Comment: Printout of this program showed that while the first 2 sequences oscillated, the third converged to 1/4. By this time, some of the students were suggesting that this process of repeated averaging might be tried for any arbitrary divergent series in the hope of eventually arriving at a sequence which would converge. In fact, what these students were doing was re-discovering Cesaro's definition of summability of series. That method assigns a number as a "sum" of a series if some average method gives rise to a convergent sequence. That is, a series $\sum a_n$ is said to be summable (C, r) to S if r is the smallest value of k such that $\lim_{n \rightarrow \infty} S_n^{(k)}$ exists, where

$$S_n^{(k)} = \left[\binom{n+k-2}{n-1} S_1 + \binom{n+k-2}{n-2} S_2 + \dots + \binom{k}{1} S_{n-1} + S_n \right] / \binom{n+k-1}{n-1}$$

Conclusion

At the conclusion of the presentation of material on infinite sequences and series, the reactions of the students were elicited. Most felt that the format of the course had demanded a great deal of work on their part. The feeling was almost unanimous that the concepts treated were clearly understood. Seventy-five percent requested that a similar format be used throughout the Intermediate Calculus course, wherever possible.

The format of the course left sufficient opportunity to try many topics, depending upon the abilities of the students in general. For example, other topics which seemed to fit nicely into the parallel lecture-assigned program structure were: methods for accelerating convergence; convergence tests; multiplication of series; and asymptotic properties of certain divergent series. The list might contain topics as innovative as the instructor might wish.

The author's impressions were not unlike those of the students. Abstract properties seemed to have been made concrete and intelligible. A great deal of genuine curiosity had been aroused. Most important of all, students were able to feel the exhilaration that resulted from discovering things for themselves.

PROGRAMS

```

A 05 LET X = 0
   10 FOR I = 1 TO 100
   15 LET Y = SQR(2+X)
   20 PRINT Y;
   25 LET X = Y
   30 NEXT I
   35 END

B 05 FOR N = 10 TO 1500 STEP 10
   10 LET X = (1+1/N)^(N+1)
   15 PRINT X
   20 NEXT N
   25 END

```

```

C 05 DIM S(500)
   10 FOR N = 1 TO 500
   15 LET S(N) = 1/N
   20 NEXT N
   25 FOR I = 495 TO 5 STEP -5
   30 LET X = 500 - I
   35 LET J = INT(I * RND(1) + X + 1)
   40 LET K = INT(I * RND(1) + X + 1)
   45 LET R = ABS(S(J) - S(K))
   50 PRINT J; "TH AND"; K; "TH TERMS:"; R
   55 NEXT I
   60 END

D 05 DIM F(101)
   10 LET F(1) = F(2) = 1
   15 FOR N = 3 TO 101
   20 LET F(N) = F(N-1) + F(N-2)
   25 NEXT N
   30 PRINT "NTH TERM", " F(N) ", " FORMULA", " RATIO"
   35 FOR I = 5 TO 100 STEP 5
   40 LET X = (((1+SQR(5))/2)I - ((1-SQR(5))/2)I) / SQR(5)
   45 LET Y = F(I) / F(I+1)
   50 PRINT I, F(I), X, Y
   55 NEXT I
   60 END

E 05 LET X = .1
   10 LET T = X
   15 LET S = X
   20 FOR I = 3 TO 1001 STEP 2
   25 LET T = ((T * X2) * (I-2)) / I
   30 IF T <= .0001 THEN 45
   35 LET S = S + T
   40 NEXT I
   45 PRINT "X =", X; "SERIES =", S; "IN"; I; "STEPS"
   50 LET X = X + .1
   55 IF X = 1 THEN 65
   60 GO TO 10
   65 END

F 05 DIM S(500), T(500), A(500)
   08 INPUT N
   10 LET X = 0
   15 LET I = J = K = 1
   20 FOR M = 1 TO 500
   25 LET S(M) = 1 / (2 * M)
   30 LET T(M) = -1 / (2 * M + 1)
   35 NEXT M

```

576

```

40 LET A(K) = S(I)
45 LET I = I+1
50 LET X = X+A(K)
55 LET K = K+1
60 IF K = 50 THEN 90
65 IF X > N THEN 75
70 GO TO 40
75 LET A(K) = T(J)
80 LET J = J+1
85 GO TO 50
90 LET C = 0
95 FOR L = 1 TO 500
100 LET C = C+A(L)
105 PRINT A(L),C
110 NEXT L
115 END

```

```

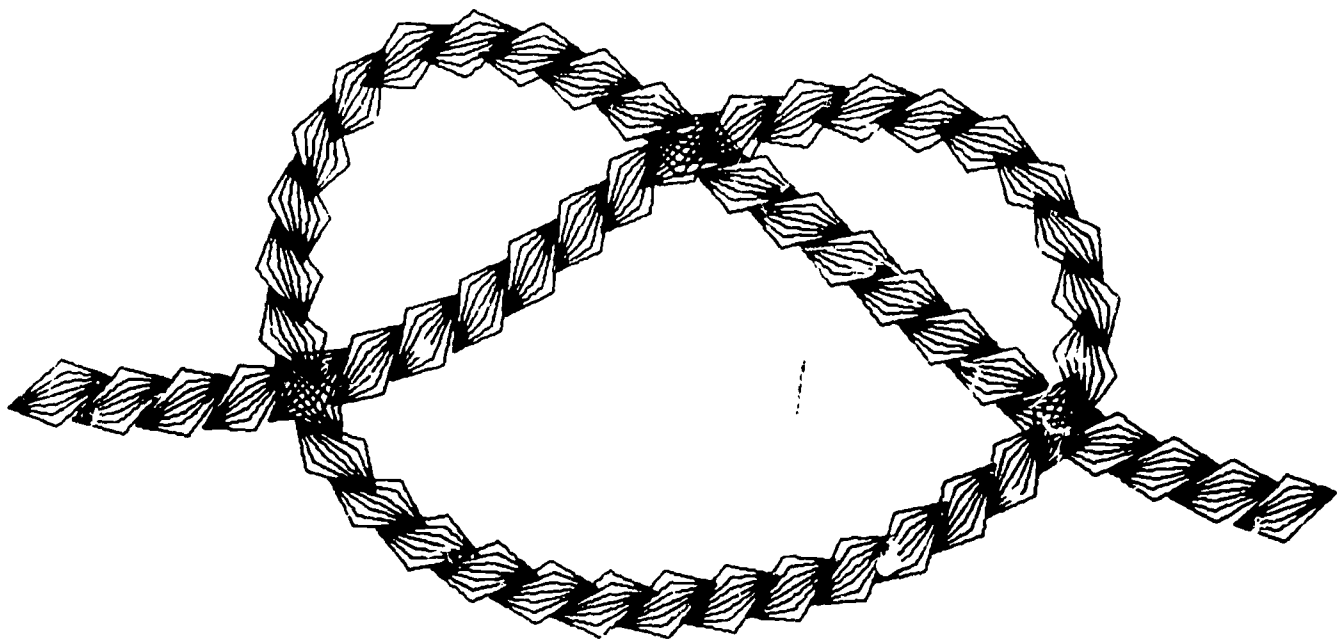
G 10 LET S = T = 0
20 PRINT " S", " S'", " S'' "
30 FOR N = 1 TO 100
40 LET S = S+1/2↑N
50 LET T = T+S
60 PRINT S, T/N
70 NEXT N
80 END

```

```

H 05 DIM S(100)
10 PRINT " S", " S'", " S'' "
15 LET K = 0
18 FOR J = 1 TO 100
20 LET K = K+(-1)↑(J-1)*J
23 LET S(J) = K
25 NEXT J
30 FOR N = 1 TO 100
35 LET A = B = C = 0
40 FOR I = 1 TO N
45 LET A = A+I*S(N+1-I)
50 LET B = B+I
55 LET C = C+S(I)
60 NEXT I
65 PRINT S(N), C/N, A/B
70 NEXT N
75 END

```



George Riske

GREAT CHAIN by George Riske
Problem: Continuous polygon forms

578

590