C
O
COMPUTERS IN
UNDERGRADUATE
SCIENCE
EDUCATION

CONFERENCE PROCEEDINGS

Commission on College Physics

CONFERENCE ON COMPUTERS IN UNDERGRADUATE SCIENCE EDUCATION

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Morning	Introductory Session	Computational Mode II	Simulation Mode	Computer - Supervised Instruction I	First Things Last
Afternoon	Computational Mode I *	Interactive Computer Graphics / Computer - Generated Film *	Analog Computers	Computer - Supervised Instruction II	The Future
Evening	Physics Workshop / In The Beginning / EDUCOM/EIN	CAI Workshop / Demonstrations	Computer Film Theater	Computer Languages Seminar	

* Concurrent Sessions

COMPUTERS IN UNDERGRADUATE SCIENCE EDUCATION

CONFERENCE PROCEEDINGS

Chicago, Illinois, August 17–21, 1970

Held under the auspices of the
Commission on College Physics
and the
Illinois Institute of Technology

Editor
RONALD BLUM

Associate Editors
Alfred M. Bork
Burton D. Fried
Donald C. Martin
Harold Weinstock
Karl L. Zinn

COMMISSION ON COLLEGE PHYSICS
COLLEGE PARK, MARYLAND
1971

Available from
American Institute of Physics
335 East 45th Street
New York, New York 10017

Library of Congress Catalogue Card Number 75-161603

FRONTISPIECE: The Program of the Conference on Computers in Undergraduate Science Education.

Foreword

The Commission on College Physics (CCP) has had a long-term interest in the computer as an instructional element. Some of the earliest experimentation in writing interactive programs in physics took place at the "Working Conference on New Instructional Materials" jointly sponsored by the CCP and the University of Washington in the summer of 1965. The following fall the CCP held a "Conference on the Uses of the Computer in Undergraduate Physics Instruction," the report of which was our first major attempt to stimulate physicists to experiment in this new medium.

There has been some experimentation during the past five years, the latest results of which ar· reported in this volume. While there have been isolated successes, there has been no massive breakthrough. The physicists, predictably, quickly turned their hand to writing computational programs for student use, and it seems clear that most physics majors now have the opportunity at some time during their career of solving a problem or reducing data on a computer. Although many of us had visions (and still do) that the great speed and flexibility of the computer would soon lead to all kinds of widespread student-computer interactions, this hope has not yet been realized. Already in Seattle we learned of the enormous cost of software development, and subsequent experience has only confirmed this painful awareness.

The idea of using the computer as a low-level tutor, always on call, which could lead a student through step-by-step learning of difficult concepts, correcting misconceptions and building understanding, is an attractive one. It is an idea put forward by many of the speakers whose thoughts are recorded in this volume. But even now, five years later, these are still only intimations of how it might be someday. We must get beyond these glimpses of the future and begin its design. This will not be accomplished all in one piece but small step by small step. We need more examples of tutorial programs whose instructional objectives are carefully spelled out and whose results are critically evaluated. We need to collect and codify those things that can be done; a checklist that says "We can do this, and this, and this . . . ."

The results of the Conference in Computers in Undergraduate Science Education reported in these *Proceedings* should move us toward this goal. In it are presented examples, techniques, criticism, and stimulus. It both reviews and previews, and it is our hope that wide distribution of this report will encourage more physicists and other scientists in the coming years to take up the experimentation of the sixties and carry it into the classrooms of the seventies.

John M. Fowler, *Director,*
Commission on College Physics

Preface

In November of 1965, the Commission on College Physics (CCP) sponsored a conference on Uses of the Computer in Undergraduate Physics Instruction, held at the University of California at Irvine. The report of the conference, *The Computer in Physics Instruction*, has been widely distributed to thousands of educators and scientists on request, and the continuing demand for it as well as the growth in computer usage played an important role in emphasizing the need for another, broader, conference, which was held August 17-21, 1970, at the Illinois Institute of Technology (IIT) in Chicago.

Since 1968, IIT had been developing a Regional Computer Network under a National Science Foundation grant, designed to serve not only IIT itself but also smaller institutions of higher learning in the Chicago area with high-quality computer facilities. An integral part of this effort was a cooperative approach to the development of computer-based, educational materials by participants from these various institutions under the direction of Group Leaders based at IIT. It was natural for Harold Weinstock of IIT, Physics Group Leader, to visit the Commission to discuss our mutual concerns, and out of these conversations grew the idea of a second "Irvine Conference," to be sponsored jointly by CCP and IIT. This plan received the encouragement of both John M. Fowler, Director of the Commission on College Physics, and Peter G. Lykos, Director of the IIT Computation Center. And so, with the support of the National Science Foundation's Office of Computing Activities, the conference was born.

When the Steering Committee* met in December 1969, the first problem was to decide upon a name for the conference. After considerable debate, the Steering Committee decided to call it the Conference on Computers in Undergraduate Science Education—Physics and Mathematics. This was a compromise between those who felt we should restrict the conference to the "purest" disciplines, physics and mathematics, and those who believed that we should strive for the mainstream of computer usage in science education, embracing chemistry, engineering, and anything else deemed sufficiently technical to qualify for the designation of "hard" science. As it turned out, however, it was in the nature of things that a conference conceived and

*Edward Adams, IBM Watson Research Center, Yorktown Heights, New York; Ronald Blum, Commission on College Physics, University of Maryland; Alfred Bork, University of California, Irvine; Herbert Peckham, Gavilan College; Edwin Taylor, MIT Education Research Center; Harold Weinstock, Illinois Institute of Technology.

initiated by physicists would be heavily biased toward physics and, secondarily, mathematics, although we tried to provide a warm welcome for contributions from other disciplines. Thus, in the end, it appeared that the conference was very aptly named. Ultimately, it became known as the COMUSÈ Conference.

This volume contains about two thirds of the papers delivered at the conference. Responsibility for selection was placed upon the Editor and the five Associate Editors. We were all conscious of the fact that, within the practicable constraints, we desired to produce a volume that would be as unified and coherent as possible under the circumstances, rather than simply a collection of diverse papers. With this in mind, we have chosen those papers contained herein to give the widest possible coverage of this new and growing field.

Clearly, we could not hope to consolidate the body of existing knowledge of the uses of computers in college science education. Such an undertaking would of necessity be (a) encyclopedic and (b) obsolete within five years. However, these *Proceedings* do include examples of almost every facet of the state of the art, and we have laid particular stress on providing information of such a nature as to enable our readers to become themselves practitioners with economy and dispatch. Thus it is hoped that this volume can provide a conceptual groundwork which will be a distinct contribution to the expanding literature of computer uses in education.

The evening sessions (see Frontispiece) were less formal and are not represented here. However, two monographs are in preparation reporting on the outcome of the workshops and are planned for publication by the Commission on College Physics. The Physics Workshop (John Robson, chairman) is expected to produce a workbook of computer-based physics problems. The CAI Workshop (Noah Sherman, chairman) has produced a program for teaching Gauss's Law in the tutorial mode based upon the use of a simplified author language presented in the Computer-Supervised Instruction session.

The conference unquestionably generated the kind of enthusiasm hoped for; people wholeheartedly joined in discussions, sat through lengthy and arduous sessions from early in the morning to late at night, and managed to fill up their remaining time by working in the workshops and meeting in informal groups in the evenings. It seems safe to say that in this "spiritual" sense of creating a feeling of community and links between interested parties, we achieved our goal. This was due in no small measure to the financial encouragement provided the attendees. First, IIT made room and board available at nominal cost; second, a gift from the IBM Corporation and another sum provided by the CCP enabled some 50 people to attend the conference who might otherwise have been unable to come. Although they rarely received as much as half of their needs, these people were typically among the most enthusiastic. All told, the conference was attended by nearly 400 participants, about 80% from smaller schools. Thus it appears that we did reach many members of our primary target population—those who have had little opportunity to use computers or to learn about them.

This conference would not have been possible without the cooperation and assistance of many people. The initial stimulus came from John M. Fowler, Director of the Commission on College Physics; Harold Weinstock, my Co-Director, found the conference a home and was instrumental in obtaining the active cooperation of IIT; Ronald Stiff and Lee Rodriguez of IIT were invaluable in attending to those vital administrative tasks essential to the smooth functioning of the conference; Arthur Melmed of the National Science Foundation's Office of Computing Activities provided indispensable comments and criticisms in the early stages of the organization of the conference as did the members of the Steering Committee; and Robin Daugherty of the CCP has been of great assistance throughout, especially during conference week, as was Carol Fennell of IIT. The conscientious and dedicated services of the Associate Editors, Alfred Bork, Burton Fried, Karl Zinn, Harold Weinstock, and Donald Martin helped to make these *Proceedings* possible; and the generous and unstinting advice of Jacqueline Boraks helped to make them legible. Finally, sincere thanks are due to the secretarial staff of the CCP for their assistance and fortitude in the face of ever-increasing demands upon their patience and stamina.

In conclusion, I sincerely feel that the COMUSE Conference was most memorable for the human links forged there and for the personal interactions that took place. I have seen some of the benefits of this already; I hope to see more in the future.

Ronald Blum
EDITOR

Contents

Introduction

RONALD BLUM

The Conference on Computers in Undergraduate Science Education—Physics and Mathematics (COMUSE) was essentially organized along the lines of computer usage, the four primary modes being (1) calculator, (2) simulator, (3) tutor, and (4) administrator. This structure is also reflected in the organization of these Proceedings. To ensure that the conference would have a definite structure and unity, fully half of the fifty speakers were selected by the Steering Committee and came as invited speakers. The remaining contributions were chosen from the large number of responses to our call for papers. The workshops and invited review papers ensured that the conference would be educational in scope *and* purpose while the other contributions described current developments.

As scientists we most naturally think first of the computational mode in which the computer performs our calculations for us. Hence, the first two full sessions were devoted to this use. Monday afternoon the session was devoted to problems in physics, including descriptions of the computer-based curriculum developments undertaken at IIT, Dartmouth, and the University of California, Irvine. Tuesday morning the session emphasized the application of the computer to the teaching of calculus, although the emphasis still remained very much upon its application to physical problems. The efforts undertaken at Florida State University and at the MIT Education Research Center were described.

Although all present seemed to agree that this mode of usage *belongs* in the curriculum and in the classroom or laboratory, it is clear that the major problem still remains the lack of widely available textual materials that use the computer. Those that are published are often only segments of courses or early exploration attempts. The vast majority of existing materials are precomputer in orientation, and seem to view the computer as a super slide-rule rather than a device of potential revolutionary significance for the *conceptual* structure of science education. Time and effort will be required for the evolution of a distinctively new style of instruction that will embrace the computer and its uses as one of its major elements.

Commission on College Physics, University of Maryland, College Park, Maryland.

In the simulation mode the computer becomes a device to simulate physical reality in the form of governing equations, laboratory experiments, or stochastic processes. In this mode the student is not necessarily aware of the nature of the computer program. He may merely input information and receive output derived by mathematical simulation of the physical situation.

In the Simulation session a broad range of problems was treated, including relativity, thermodynamics, optics and modern physics. The emphasis in all of them was on having the student perform "experiments" on the computer, and in every case the output was either via a graphics terminal or would have lent itself particularly well to a graphical treatment had a suitable terminal been available. This is not surprising, since most of our richest impressions are formed from pictures, not digits, and graphic devices eliminate the intermediate step of "digitizing" our intuitions, giving us information in a form that most closely simulates reality as we perceive it. For this reason, the Simulation Mode session was preceded by the sessions in interactive graphics and computer-generated film. By "interactive" we mean those devices with which the student can communicate immediately and directly, with response times on the order of seconds.

Immediately following we presented the Analog Computing session, since the analog device is intrinsically an interactive simulator, its behavior closely resembling the phenomena it is intended to describe. The analog computer contains many different electrical elements which simulate the performance of different mathematical operations, such as adding, subtracting, multiplying, dividing, differentiating and integrating voltages input to them. Furthermore, the student can more readily employ his physical intuition just as in the laboratory, continuously varying parameters by adjusting dials and knobs on the computer, and immediately observing their effect on the output. Again, he does not need to digitize his feelings into numbers and symbols.

The analog computer has been largely neglected by physicists, somewhat less so by engineers, in favor of the more versatile, widespread and glamorous digital computer. When planning the conference it was only allotted half an afternoon. However, the response was surprising, and so Analog Computing was expanded to an entire afternoon at the expense of the Computer-Generated Film session. This is not to imply a lack of interest in film, since a four-hour Theater on Wednesday evening was approximately evenly divided between computer-animated films and films about interactive graphics systems.

Applications of computer graphics were so widespread as to make it difficult to categorize them. To further point up the "naturalness" of graphic simulation as an instructional tool I would also add the observation that the majority of the real-time demonstrations that took place at the conference occurred in the three sessions just discussed.

Rather than further proliferate the alphabet soup that customarily goes with the tutorial-conversational mode (CAI, CAL, CMI, etc.), we decided to call the next two sessions Computer-Supervised Instruction but refrained from abbreviating it. Under this rubric we also included the administrative mode in which the computer can give exams, grade them, and make individual assessments and recommendations concerning a student's progress.

In the tutorial mode, the computer is programmed to converse with the student. It can give him information, ask him questions and react to his responses through a typewriter, CRT or an ordinary time-sharing teletype console. Or it can respond to the student's questions. Many different strategies are possible, varying according to the degree of complexity of the feedback and branching which the program undertakes as the result of the student's responses.

Among others, these sessions heard reports on the highly developed PLATO system at the University of Illinois, Urbana; the Florida State University experimental physical sciences course, which was completely computer-administered and computer-managed; and the multimedia U.S. Naval Academy experimental introductory physics course. Unfortunately, what emerges is the inescapable fact that these systems are still much too expensive for any but developers and experimenters to use; due, in large part, to the complexity of the software.

On the last day of the conference we devoted the morning session to the practical administrative details of computer usage and funding on campus, with particular reference to education. A troika of speakers delivered short papers on The Politics of Local Persuasion, discussing the tactics of obtaining and using a computer in, respectively, a university (IIT), a liberal arts college (Vassar), and a junior college (Gavilan). The pitfalls, such as administrative data processing were discussed as well as ways of wearing down the opposition. Useful tactics apparently involve perseverance and a gift of gab, while technical knowledge of computers seemed to be largely irrelevant, at least at the beginning. Though hardly a prescription for success, this session at least provided a context for people who are thinking about acquiring facilities. It also generated a discussion almost as long as the three papers themselves. Another paper included here also discussed in detail the economics and pragmatics of campus-wide computing at Stanford University, one of the major installations in the nation.

The last session of the conference was very frankly speculative, the major emphasis being on the probable thrust of new technical developments, the increasing application of technology to education (felt, with some reservations, to be desirable), and the probable direction of Federal funding and encouragement in the future as our society moves more and more from one based upon industrial production to one based upon educated manpower.

It became clear from the many requests from participants for information of a very detailed nature–bibliographies, prices, hardware specifications, software packages, names, addresses, etc.–that many are planning (or hoping) to add computers to their teaching repertoire. Therefore, we have also included some modest appendixes which will provide some basic information, as well as a list of people who contributed to the COMUSE conference in order to encourage communication. Interested parties are urged to write the Editor and authors with their suggestions and comments.

I

COMPUTATIONAL MODE

Introduction

Half the papers in this section are the result of large-scale efforts to improve physics and mathematics instruction by making available computer-based materials in the computational mode and/or integrating them into the traditional course structure. These papers, by Weinstock, Bork, Huggins, Schey and Stenberg, are not radical in terms of their subject matter but in their approach to it through computer-based methods. In each case numerous concrete examples are given.

Weinstock discusses the *modus operandi* of IIT's curriculum development venture designed to collect and develop computer-based materials directed toward the solution of specific classical problems which occur in science education, giving as a detailed example a rocket landing on the moon, one which has many characteristics of a simulation. Bork's paper describes the organization of a computer-oriented course in introductory mechanics, including numerous sample problems and a tutorial man-machine dialogue on kinetic energy. Huggins' strobe-computer laboratory is part of the effort on Dartmouth's Project COEXIST and lavishly illustrates the complementarity between data collection, computer usage (via time-shared teletypes) and the concepts of mechanics.

Schey and Stenberg both discuss the development of computer-oriented calculus courses, and both lay heavy stress upon the use of the computer to demonstrate the · 'ure of limits. Despite this strong similarity both are included for the sake of comparison and because they represent ambitious ventures, but from two different points of view. The inspiration for Schey's paper is clearly drawn from the physics community, while Stenberg's can be ascribed to the mathematician's point of view. Hence these two articles provide an interesting counterpoint, but it may be noted that, with the exception of quadrature, little is said about computer-based *methods*.

Bowers' paper describes a one-semester general physics course in which twelve computer-based exercises have been incorporated into the laboratories while Goodman and Gauvin give computer applications to specific problems in physics, the former using a desk-top computer and the latter a time-shared teletype with the APL language. The paper by Burnstein, Swanson and Veirs is, properly speaking, neither physics nor mathematics but statistics. However, it should be of considerable interest to physicists since it describes programs for data reduction used in a senior physics laboratory.

Goodman and Stoner (see Section III, Simulation Mode) are the only papers in this volume in which the desk-top computer is explicitly mentioned. These programmable calculators look and act, in many respects, like ordinary calculating machines. However, they are really small computers containing function keys which compute sines, cosines, logarithms, etc., as well as keys for digits and arithmetic operations. They have the ability to "learn" and store short programs, which may be read from tape cassettes, magnetized cards, or punched cards, and they may also include provisions for accessories such as printed output and *XY* plotters. Costing typically of the order of $5,000–10,000, these machines offer considerable convenience and economy for most classroom uses, although they lack, as yet, the ability to handle subscripted variables and arrays. The type of programming required for such machines is distinctive, and a sample program has been included in the paper by Goodman.

Finally, there is the paper by Engelman which, if you have not previously encountered algebraic symbol manipulation programs, is bound to astonish you. The very existence of programs that can multiply polynomials, decompose rational functions into partial fractions and perform other symbolic tasks gave rise to one of the most provocative statements to come out of the conference. That is, where Engelman states: "If we are correct in our contention that the practice of applied mathematics is about to change radically, then we must question whether it is not possible that we are teaching the wrong skills." Considering the fact that almost all the problems treated in this section are elementary ones, this is clearly not an immediate and pressing question in connection with the "mechanization" of mathematics. However, the broad implications of this statement should be kept in mind by all science educators in connection with computer usage generally.

A Report on a Cooperative Venture in Curriculum Development

HAROLD WEINSTOCK

THE PROJECT

A two-year project was sponsored by the National Science Foundation (NSF) for the purpose of promoting the use of computers in college curricula. The program involved the cooperation of twelve colleges and universities in the midwest area, all of which used the facilities of the Illinois Institute of Technology (IIT) Computation Center through remote access teletype units. The regional computer network established under this National Science Foundation program is one of ten originally funded for this purpose. However, to the best of my knowledge, it is the only one of this original group which has stressed the development of new curricula from the beginning.

The institutions involved with IIT in this project vary in size and in distance from IIT. Most of them are small four-year liberal arts colleges, others are two-year junior colleges, and one of the institutions is a large university. While many of the cooperating institutions are found in the Chicago metropolitan area, four of the institutions are more than 200 miles away. The development of curricula has been carried out in the following disciplines: Mathematics, Physics, Chemistry, Biology, Business and Economics, and Education-Psychology-Sociology, although well into the project Sociology was formed into a separate study group. An informal linguistics group has also associated itself with this venture.

The work in each discipline is guided by a Group Leader. Each Group Leader has been accorded 25 percent teaching release time during the academic year to devote his efforts to the project, and also has been supported during part of each of the three summers encompassing the project. Approximately one half of the Group Leaders are members of the IIT faculty, while the remainer have been recruited from other colleges in the Chicago area. Group Leaders have introduced suitable materials from the "outside world," coordinated the activities of representatives of the member

Department of Physics, Illinois Institute of Technology, Chicago, Illinois

colleges in their disciplines and led in the development of new curriculum materials. The resources of the IIT Computation Center and the freedom to explore new techniques of teaching gave the Group Leaders a tremendous opportunity to become involved in what must be considered the infancy of this pedagogical technique.

Each Group Leader had the services of an undergraduate programming assistant, and such assistants were sometimes also available in participating institutions. These undergraduates represented one of the most valuable resources in the project, providing useful criticisms and comments as well as a much needed service which freed Group Leaders and faculty participants to consider more fully the pedagogical applications and implications of this work. It also goes without saying that the students were generally much quicker in picking up errors and variations in computer usage and in doing actual programming.

The Group Leaders enjoyed greater advantages than their counterparts at the participating colleges who, in addition to teaching as many as six to eight courses at one time, were also endeavoring to master the uses of the computer in their curricula. The latter were generally given no time off for such development and had only the use of one or two on-line teletype units to the IIT Computation Center. They also maintained contact with their particular Group Leaders and other people at the IIT Computation Center, but these contacts were somewhat limited. However, these hard-working teachers, coordinated through a Principal Investigator at each institution, were the backbone of this project.

Direct communication to all participants was made initially at a three-week meeting on the IIT campus during the summer of 1968. Subsequent meetings of this type took place for one and one-half days each in the winter and spring of the 1968-69 academic year, for one week in the summer of 1969 and for one and one-half days in the winter of the 1969-70 academic year. The majority of time during these meetings was occupied along disciplinary lines. However, there were several joint interdisciplinary sessions which featured general use of the computer system and the airing of various problems common to many of the disciplines. The joint sessions were also used for discipline reports which were felt to be useful to people outside a given discipline. Often many revelations took place during such reports, and it was found that a particular exercise in one discipline could in fact be quite applicable to another.

In the initial three-week meeting all participants were given a one-week crash course on programming and the use of the IIT Computation Center through teletype access. For those who already had some knowledge of computer programming, the crash program proved reasonably successful, but for those with no previous experience it was rather traumatic. Nevertheless, such people had ample opportunity to develop their programming and contact skills with the computer in the ensuing two weeks at IIT and later on in the summer, prior to the start of the academic year (although in many institutions delay in delivery and installation of teletypes precluded such activity). Not all participating colleges had representatives in all disciplines covered in the project; in some disciplinary groups there might be as few as five institutional representatives in addition to the Group Leader, while in the area of Mathematics, almost every institution was well represented.

Attempts were made in each disciplinary group to have each participant cover a particular area in curriculum development, either within some very general course common to all the institutions or in a variety of courses. This latter technique worked best with the Mathematics Group, but not so well for some of the other disciplines where there was unevenness in the over-all curriculum. Of course, one might also suggest that the mathematicians have the greatest need and ability to use the computer in curriculum efforts, but one should temper this with the fact that previous to the inception of this project, other mathematical efforts along these lines appeared to be rare.

The languages used for communication along the regional network were initially FORTRAN IV and IITRAN, a local language with less formal error statements, formating, etc., than FORTRAN IV and which allows for batch processing from remote locations. More recently, with IIT's new Univac 1108 system BASIC has been made available. In addition, Professor Robert Dewar of the IIT Computation Center developed a new language which he called CALCTRAN, basically a short form of conversational IITRAN with limited programming capabilities. The initial motivation for the use of CALCTRAN was to enable computer users to have the equivalent of an electronic desk calculator for computation. However, the programming capability of up to twenty statements proved a boon to those with simple problems which could best be handled in a conversational manner.

The conventional FORTRAN IV and IITRAN had to be used non-conversationally due to the overtaxed IBM 360/40 system in use at the start of this project. To be sure, batch processing and non-conversational responsiveness are drawbacks to the use of a remote-access facility, but we have been able to show many useful developments under such limited conditions. The addition of the CALCTRAN capability gave an even greater facility to the network. Such a system is relatively inexpensive when compared with the cost of an interactive system for all languages used. The CALCTRAN in particular, while conversational, is in fact a rather inexpensive language to use. In physics, the one-dimensional quantum-mechanical square-well problem, a Monte Carlo model for radioactive decay, and a simulated moon landing were quickly programmed and executed in CALCTRAN.

CURRICULUM DEVELOPMENT

Day-to-day communication between participants in the Network was accomplished by means of the Cooperative Program Exchange Service (COPES). When a participant developed curricular material which could be of use to other participants, usually within his own discipline, he sent it to the Project Administrator at IIT for distribution. The COPES entry was usually in one of several forms:

(1) a brief abstract explaining the use of an accompanying program listing and (sometimes) flow chart;
(2) a review of some article or application developed outside the Network but adaptable for local use;
(3) a comprehensive document called a Computer Enriched Teaching Unit (CETU), which presents a complete educational application.

The contents of the COPES library represents one of the most important achievements of this project, one that will have some lasting significance. In many ways the COPES library resembles that which has been started by the Commission on College Physics in cooperation with the American Journal of Physics and which is known as the Computer Library for Instruction in Physics (CLIP). The currently reported project and the Commission on College Physics are both nearing extinction, but one hopes that some arrangement can be made to continue such efforts so that the work of the past will not be wasted. With other networks generating libraries similar to COPES and CLIP, there will be much needless duplication; it would be most useful to have a single repository with easy access to the entire academic community.

There is a tremendous inertia in using the computer as an educational tool. This problem is further compounded at most Network campuses, where the glamour of having a "real live" computer, programming courses, and bona fide computer experts on campus is missing. All that one does have (in most cases) is a single on-line model 33 teletype unit and one or two off-line units. It became imperative, therefore, for the Group Leaders and participating faculty at the Network institutions to introduce computer applications that would be appropriate and easy to use. This led to the generation of a number of so-called "canned" programs which served the purpose of getting students to access the computer and showed them that it could improve their understanding in a variety of subject areas.

Once initiated in this relatively effortless way, both faculty and students—especially the students—are ripe for either formal or informal programming instructions. At first, a member of the IIT Computation Center staff did yeoman work in presenting informal programming and teletype operation instruction to a number of Network campuses. Now, most of these same campuses are offering "for-credit" programming courses using local resources.

One example of a particularly effective CETU is a completely self-contained unit on numerical integration authored by Professor P. Chiarulli of IIT and D. Venhaus, an undergraduate. This unit can be used at different levels but could most easily be introduced into a first-year calculus course. It is not tied to any particular text and requires only a minimum of class time for introduction to the unit itself. A student is not required to have any prior knowledge of computer programming, but it is assumed that the concept of the integral as the limit of a "Riemann sum" has been covered in the course textbook. However, the theory of numerical integration by the Trapezoidal Rule and Simpson's Rule is included in the CETU documentation.

In the Theory and Analysis section of the CETU, a review is presented of the Rectangular Approximation, the Trapezoidal Approximation and the Simpson's (or Parabolic) Rule Approximation for numerical integration. With this as background, a section follows on the input necessary for the computer, and a specific example is given for a typical function. Following this detailed explanation, a problem set is presented, part of which deals with the effective application of the method to problem solving; while another part, for students with programming experience, involves the nature of the algorithm itself.

The statements that a student must punch out on his tape are broken down into three areas:

Preliminary Control Instructions (PCI)
Job Assembly Instructions (JAI)
Terminating Control Instructions (TCI)

These are indicated in the numerical integration program of Figure 1. Note that on the last JAI line, the numbers 0, 1, 10, 200, 0, 0, 0 appear. The CETU

```
PCI      $IITRAN CHIARULLI, P. SAMPLE PRØBLEM
         ./ INCLUDE PRJ.NSFMA044

         Y←9*X↑8 - 8*X↑7 + 7*X↑6
         GO TØ RETURN
         END NSFMA044

JAI      EXEC NSFMA044
         'Y = 9*X↑8 - 8*X↑7 + 7*X↑6'
         'RECT TRAP SIMP'
         0 1 10 200 0 0 0

TCI      $QUIT
```

FIGURE 1 Teletype tape for numerical integration program.

states that the first two numbers define the interval to be integrated and that the next number represents the number of subintervals. The number after that gives another number of subintervals for the same limits of integration, and then finally the three zeros represent a stop to the program. The student is also informed of the variations that he can employ and the problems that may result if he fails to input correct data. Figure 2 shows typical output; the integrand is shown first, followed by the limits of integration. Then the result of the quadrature is given for the three different methods, for each of the two subintervals chosen, i.e., $N = 10$ and 200. The program is organized to "forgive" errors whenever possible but presents error messages to the student.

```
Y = 9*X↑8  -  8*X↑7  +  7*X↑6


A = 0.              B = 1.00000     N = 10
    I(RECT) =        .64805
    I(TRAP) =       1.04807
    I(SIMP) =       1.00122


A = 0.              B = 1.00000     N = 200
    I(RECT) =        .98012
    I(TRAP) =       1.00012
    I(SIMP) =       1.00000
```

FIGURE 2 Output for the integration indicated in Figure 1.

The above unit can be particularly useful in the early part of a semester in helping students to become familiar with computer terminals as well as in helping them to understand what may, for many, be a difficult concept, that of the integral. Once students have learned what a computer can do for them, they can be exposed to other problems that require some elementary and readily attained programming skills. Since Simpson's Rule is a very widely used and accurate technique of quadrature, it should also be of value to the student in his later work.

In Physics, our first successful and widely used CETU presentation involved another "canned" program, one which performed the simple operation of adding a total of up to twenty sine waves and presented the output in either tabular form or graphically via asterisks, or both. The student input involves choosing five parameters for each wave: amplitude, wavelength, frequency, phase constant and direction of propagation; the variable to be held constant, either position or time; and the form of the output. The instructional part of this unit discusses several applications within the subject of wave motion, e.g.. standing waves, beats, Fourier synthesis and propagation in a dispersive medium, which are amenable to the program in question. A computer application of this type is hardly exotic, yet it can be adapted to the most modest computing system; and it opens up a class of problems which previously could not otherwise be investigated directly by students in any reasonable length of time.

By far the most successful physics application in terms of generating immediate student interest is a conversational lunar landing simulation. The earliest version of this was a modest and basically unrealistic simulation inasmuch as it assumed a flat moon, constant LEM (Lunar Excursion Module) weight and gravitational acceleration, irrespective of fuel expended or height above the moon's surface, and only vertical motion. The student is told that the LEM weighs ten thousand pounds on the moon, that its initial altitude is one thousand feet above the moon and that the initial velocity is one hundred feet per second downward. He is then told to input a number that will represent the reverse thrust (in thousands of pounds) to be exerted for each of the next three seconds. The program then calculates and prints the new position, velocity and amount of fuel left after each one-second interval.

After each computation, in addition to the printout values of altitude, velocity and fuel supply, the program print out:

(1) TOUCH DOWN, if the altitude is less than ten feet and the velocity is less than ten feet per second;
(2) CRASH, if the altitude is less than ten feet but the velocity is ten feet per second or greater;
(3) OUT OF FUEL, if the initial fuel supply has been completely expended and the calculation proceeds to the next interval, with zero thrust.

If none of these three situations occurs after three seconds, then the student is requested to input a new value of thrust for the succeeding three seconds. This routine may be continued until a TOUCH DOWN or CRASH occurs..

A later version of the lunar landing simulation proved more realistic by considering the problem a two-dimensional one. The "pilot" not only had to land softly, i.e., slowly, enough on the surface, but he had to direct the LEM to a point *not* directly beneath the position at which he took control. In this improved version the "pilot" controlled three parameters: the angle of reverse thrust, the magnitude of his thrust and its duration. The initial conditions were given as $x = y = 100{,}000$ ft, $v_x = v_y = 10{,}000$ ft/sec, and a fuel supply of 2,000 lbs. It proved challenging to be able to land on target and safely ($x = y < 10$ ft, $v_x = v_y < 10$ ft/sec) with a limited fuel supply. Once this had been accomplished, the goal became to do so with a minimum expenditure of fuel. Students soon learned the advantage of allowing the LEM to "drift" under the moon's gravitational influence.

Most recently, this program has been upgraded and adapted to conversational BASIC. The most significant new feature is that a spherical shape has been assumed for the moon (Figure 3), and realistic data from the Apollo XI flight and corrections for the change in gravitational acceleration with height above the moon's surface have been included. The BASIC program and the statement of the problem as presented to the student are included in the Appendix to this paper; the flowcharts are shown there in (Figures 4 and 5). It should be noted that the calculations in this problem are carried out in an inertial reference frame. For advanced students, an interesting exercise would be to have them rewrite the program in terms of a rotating reference frame, since the effective acceleration experienced by the "pilot" is in this frame. To relate the inertial acceleration to the effective acceleration, the centripetal, Coriolis and angular acceleration terms would have to be included.

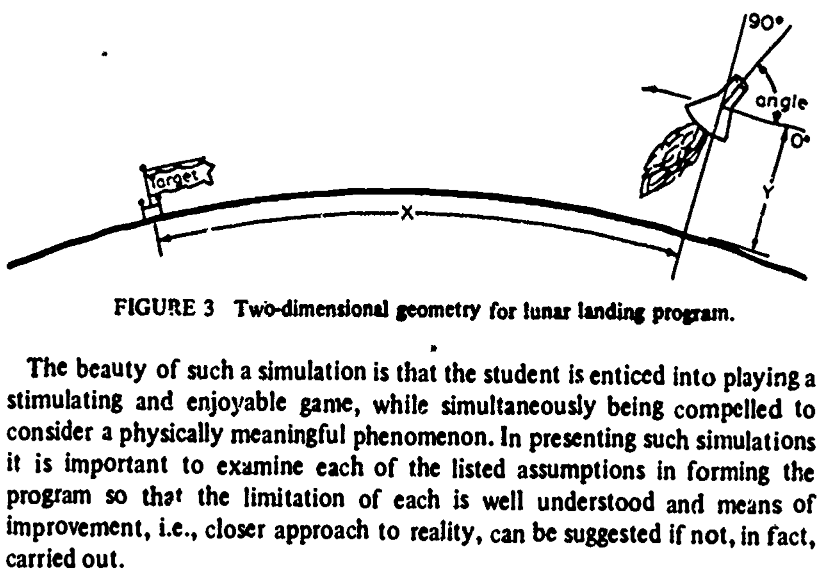


FIGURE 3 Two-dimensional geometry for lunar landing program.

The beauty of such a simulation is that the student is enticed into playing a stimulating and enjoyable game, while simultaneously being compelled to consider a physically meaningful phenomenon. In presenting such simulations it is important to examine each of the listed assumptions in forming the program so that the limitation of each is well understood and means of improvement, i.e., closer approach to reality, can be suggested if not, in fact, carried out.

CONCLUSIONS

This curriculum development project can be considered only a qualified success at best. At most of the Network colleges only a small fraction of the materials developed have been used, and only by a small fraction of the faculty involved, in a limited number of their classes. Furthermore, most of the available materials were produced through the efforts of the Group Leaders.

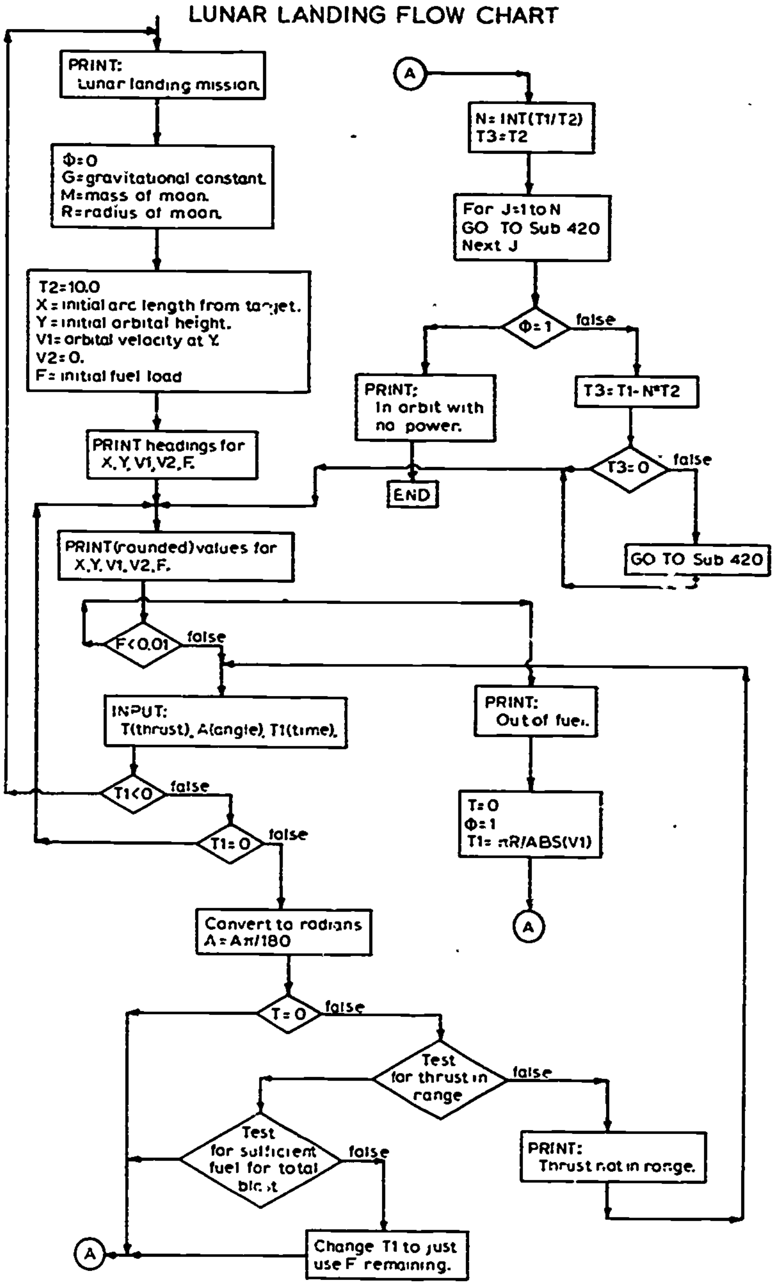


FIGURE 4 Lunar landing program flowchart.

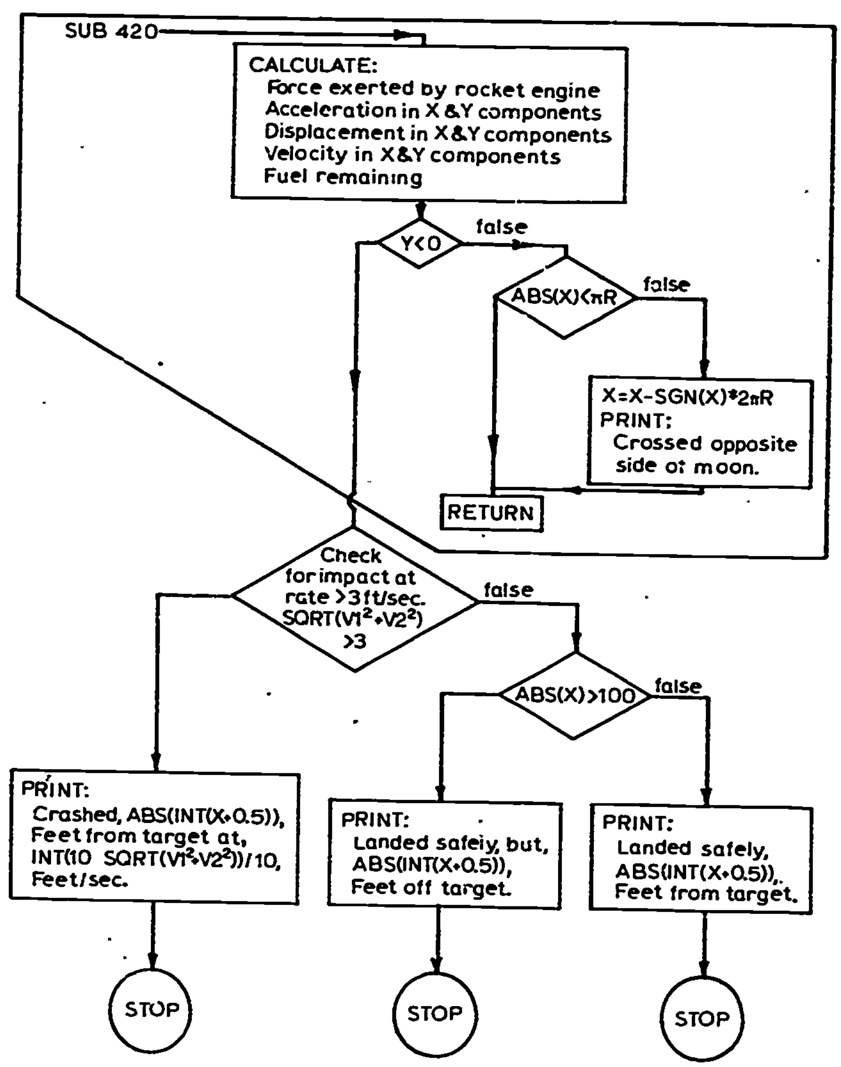


FIGURE 5 Flowchart for Subroutine 420 of the lunar landing program.

Earlier in this discussion I have referred to the participating faculty at Network institutions as the backbone of the project. Frankly, it is a weak backbone because most of the faculty involved were given no respite from staggering teaching loads and other academic duties, while a significant number were "involved" as a result of being "appointed." It is not possible to force a faculty member to engage in pedagogical development with successful results; certainly no more possible than it is to force the same individual to do scientific research. A much better way to introduce computer uses is to seek out those who have a genuine interest in this pursuit. No matter what their teaching loads and the state of local computer facilities, such individuals are the ones who will make the greatest use of computer-based teaching materials and applications, and somehow they will find the time to present these to their students.

A more productive formal approach may be to focus attention on teacher training through summer institutes. During a recent NSF-sponsored Summer Institute at IIT more usuable material was developed in four weeks by twenty-three participants than in the preceding two years of the Network project. However, even with well-trained and dedicated faculty members there is yet much opposition to new trends in many colleges, and one should stress again the importance of the enthusiasm and ability of the undergraduate students. Since educating them is our goal, a direct involvement of the students will perhaps bear the greatest fruits and do wonders toward developing new materials and generating new ideas. This project has brought a number of institutions into the "Computer Age" for the first time. It has enabled them to use some relatively inexpensive educational applications of the computer and appraised them of a multitude of other such applications. Progress has been made, however slow it may be. Finally, this project has resulted in the development of curriculum materials in several academic disciplines which are exportable with possibly some modification to almost any computer facility.* It is our hope that some of what we have done will find its way into the undergraduate curricula of the not too distant future.

APPENDIX. LUNAR LANDING

The following text and listing are shown as they appear to the student on the teletype output.

LUNAR LANDING

WOULD YOU LIKE A DESCRIPTION OF THIS GAME (1 = YES, 0 = NO)? 1

YOU ARE IN AN APPOLLO XI LUNAR MODULE WEIGHING (ON EARTH) 14,000 LB AND CARRYING 18,000 LB OF FUEL. YOU ARE IN A CIRCULAR ORBIT LOCATED IN THE MOON'S EQUATORIAL PLANE AT AN ALTITUDE OF APPROX. 68 MI.

YOUR TARGET LIES ON THE LUNAR EQUATOR A DISTANCE OF APPROX. 760 MI (GROUND DISTANCE) AHEAD. AS YOU PROCEED THE X VARIABLE WILL REPRESENT THE GROUND DISTANCE TO YOUR TARGET FROM A POINT DIRECTLY BELOW YOU ON THE LUNAR SURFACE, MEASURED POSITIVELY TOWARD YOUR INITIAL POSITION. THE Y VARIABLE WILL REPRESENT YOUR ALTITUDE.

THE X VELOCITY IS YOUR GROUND VELOCITY MEASURED POSITIVELY IN YOUR INITIAL DIRECTION WHILE THE Y VELOCITY IS YOUR RATE OF DESCENT IN ALTITUDE.

*Full documentation is available at cost upon request by writing to the author.

THE VARIABLES AT YOUR CONTROL ARE THRUST, ANGLE (OF THRUST), AND TIME (DURATION OF THRUST).

THE THRUST CAN TAKE ON ONLY CERTAIN RESTRICTED VALUES AS IN THE CASE OF THE APOLLO XI LUNAR MODULE. THESE VALUES ARE: ZERO, 1000 THRU 6000 LBS. OR 10000 LBS.

THE ANGLE CAN BE SET AT ANY VALUE FROM 0 TO 360 DEGREES. A ZERO ANGLE REPRESENTS A TANGENTIAL THRUST IN THE INITIAL BACKWARD DIRECTION WHILE A 90 DEGREE ANGLE REPRESENTS AN UPWARD THRUST DIRECTION.

THE TIME VARIABLE SETS THE LENGTH OF TIME THAT THIS THRUST VALUE IS TO BE MAINTAINED BY THE ROCKET ENGINE. NEW VALUES OF POSITION AND VELOCITY ARE CALCULATED AS OF THE END OF THIS TIME INTERVAL AND PRESENTED FOR YOU TO DECIDE ON FURTHER GUIDANCE VIA THESE CONTROL VARIABLES. THE AMOUNT OF FUEL REMAINING IS ALSO PRESENTED AT THIS TIME AS A FACTOR FOR CONSIDERATION. YOUR SPACE VEHICLE CONSUMES FUEL AT A RATE OF ONE LB OF FUEL PER SECOND PER 283.74 LB OF THRUST.

IF YOU SET THE TIME TO -1 THE PROGRAM WILL ABORT AND RESET TO ITS INITIAL CONDITIONS.

IF A CRASH OCCURS, THE EVENT WILL BE RECORDED ALONG WITH DATA ON SPEED AND DISTANCE FROM TARGET AT THE TIME IMPACT. IF YOU RUN OUT OF FUEL, THIS WILL BE RECORDED AFTER WHICH YOU MAY EITHER CRASH OR GO INTO ORBIT WITH NO NAVIGATING POWER.

THE MODULE IS CONSIDERED TO BE LANDED SAFELY IF IT REACHES THE LUNAR SURFACE (Y = 0) WITH SPEED LESS THAN 3 FT/SEC. THE LANDING MUST BE WITHIN 100 FT OF TARGET BEFORE THE MISSION IS CONSIDERED TO BE SUCCESSFUL.

YOU ARE NOW AT THE CONTROLS. HAPPY LANDING!

Program listing for Lunar Landing:

```
10 IF T1=0 THEN 20
12 PRINT
14 PRINT
16 PRINT TAB(25);"NEW START"
18 IF T1<0 THEN 50
20 PRINT TAB(25);"LUNAR LANDING MISSION"
48 LET Ø = 0
50 LET G=3.438*10↑(-8)
60 LET M=5.058*10↑21
70 LET R=5.702*10↑6
80 LET T2=10
85 PRINT
98 PRINT
100 LET X=4*10 ↑ 6
110 LET Y=3.6*10 ↑ 5
120 LET V1=SQR(G*M/(R+Y))*R/(R+Y)
130 LET F=18000
140 LET V2=0
150 PRINT TAB(3);"X";TAB(13);"Y";TAB(22);"X VEL";
```

```
155 PRINT TAB(32);"Y VEL";TAB(42);"FUEL";
156 PRINT TAB(2);"(FT)";TAB(12);"(FT)";TAB(21);"(FT/SEC)";
157 PRINT TAB(31);"(FT/SEC)";TAB(42);"(LB)"
159 PRINT TAB(2);INT(X+.5);TAB(12);INT(Y+5);
162 PRINT TAB(20);INT(10*V1+.5)/10;TAB(30);INT(10*V2+.5)/10;
165 PRINT TAB(60);INT(F+.5)
180 IF F <.01 THEN 520
190 PRINT "THRUST (LB)";
192 INPUT T
194 PRINT "ANGLE(DEG)";
196 INPUT A
198 PRINT "TIME(SEC)";
200 INPUT T1
205 IF T1 < 0 THEN 12
210 IF T1=0 THEN 159
220 LET A=A*3.1415927/180
250 IF T=0 THEN 330
260 IF T<1000 THEN 290
270 IF T<=6000 THEN 310
280 If T=10000 THEN 310
290 PRINT "THRUST OUT OF RANGE"
300 GØ TØ 190
310 IF F -T*T1/283.74>=0 THEN 330
320 LET T1=283.74*F/T
330 LET N=INT(T1/T2)
340 LET T3=T2
350 FO R J=1 TO N
360 GO SUB 420
370 NEXT J
375 IF Ø =1 THEN 640
380 LET T3=T1-N*T2
385 IF T3=0 THEN 159
390 GØ SUB 420
400 GØ TØ 159
420 LET F1=T*T3/283.74
430 LET A0=32.18*T/(14000+F)
440 LET A1=A0*COS(A)*R/(R+Y)
447 LET A1=A1-2*V1*V2/(R+Y)
450 LET A2=A0*SIN(A)-G*M/(R+Y) ↑ 2+(R+Y)(V1/R) ↑ 2
460 LET X=X-V1*T3+A1*T3↑2/2
470 LET V1=V1-A1*T3
480 LET Y=Y-V2*T3+A2*T3↑2/2
490 LET V2=V2-A2*T3
500 LET F=F-F1
504 IF Y < 0 THEN 550
508 IF ABS(X) < 3.14159*R THEN 516
512 LET X=X-SGN(X)*2*3.14159*R
513 PRINT "CROSSED OPPOSITE SIDE OF MOON"
516 RETURN
520 PRINT "OUT OF FUEL"
```

```
530 LET T=0
535 LET Ø=1
540 LET T1=3.14159*R/ABS(V1)
545 GØ TØ 330
550 PRINT
560 IF SQR(V1↑2+V2↑2) > 3 THEN 620
570 IF ABS(X)>100 THEN 600
580 PRINT "LANDED SAFELY";ABS(INT(X+.5));"FEET FROM TARGET.";
585 PRINT "CONGRATULATIONS"
590 STØP
600 PRINT "LANDED SAFELY";ABS(INT(X+.5));"OFF TARGET.";
605 PRINT "MISSION UNSUCCESSFUL."
610 STØP
620 PRINT "CRASHED";ABS(INT(X+.5));"FEET FROM TARGET AT";
625 PRINT INT(10*SQR(V1↑2+V2↑2))/10;"FT/SEC"
630 STØP
640 PRINT "IN ORBIT WITH NO NAVIGATING POWER"
900 END
```

Computer-Based Mechanics

ALFRED M. BORK

This paper discusses several ways in which the computer has been used to teach beginning Mechanics during the last two years at the University of California, Irvine. The several approaches outlined here differ widely, but all use similar computers. This usage is generally advantageous[1] in physics, particularly in mechanics. The science student should see the computer, very early in his career, in the context of a subject-matter area. He should learn the strengths and weaknesses of computer-based techniques, as compared with other problem-solving methods, and should progress with the computer as part of his available equipment. An undergraduate who does not use computers in physics is like one who does not employ voltmeters and oscilloscopes in laboratories.

A second reason for computers in physics classes is the inherent value of learning to write and debug programs, a process that obliges students to organize their thoughts. Professor Gerald Holton, Harvard University, has observed that writing a program puts the student in the role of a *teacher* explaining how to carry out a calculation to a "student," skilled in calculating, who accepts instructions very literally. The student-teacher must express reasonably precise and unambiguous ideas to make the computer "understand." In an interactive system the computer complains when the student makes mistakes, focusing his attention upon the adequacy of what he is doing. Teaching is a valuable experience open only to a few undergraduates in large universities; whereas successfully writing computer programs, i.e., teaching the computer a particular calculation, is a valuable educational experience available to many.

A third reason for computers is motivational: for many students the computer is an exciting device, which makes a course more interesting. This can be a two-edged sword—the thoroughly fascinated student may leave physics! But by and large the computer is a positive factor in motivating students.

Within Mechanics a clear goal is obtainable through using computers in the

Department of Physics, University of California, Irvine, California. Supported in part by the National Science Foundation.

computational mode, which is difficult for many students to reach without the computer. A beginning course using the computer can allow the student to see Newton's Second Law as *differential equations*, in contrast to the usual approach which is algebraic; although velocity and acceleration may be defined with derivatives, when the student *uses* the laws of motion, in lecture and problems, he treats them as algebraic statements. Computer methods allow students to work immediately with the laws of motion as differential equations, first numerically, and then analytically. We can provide students with the tools to find how *everything* moves, using the laws of motion. The algebraic approach only enables him to determine how a very limited class of objects move.

APPROACHES

Everyday Experience

Perhaps the simplest approach was used in high school with no problems. Practically everyone is familiar with cars and can make simple calculations based on driving in cars. Thus everybody knows that a car going 30 miles/hour travels 90 miles in three hours, the starting point for this treatment. This car-experience points to a way to compute the new position; the student adds to the old position the product of velocity and the elapsed time:

$$X' = X + V*DT$$

This is done first for constant velocity, a familiar situation. Then the major issue is how velocities change; we introduce Newton's Second Law, with force changing velocity. It is natural to use the same type of relation for the change of velocity as used for change of position, so the statement of Newton's Second Law looks like the statement above:

$$V' = V + F*DT/M$$

Computer programming is intermixed with the above discussion, little by little; the relations are not written in ordinary algebraic form but in a form compatible with computer language.

Standard Mechanics

A second approach is to tack a week's work onto a standard beginning course, assuming that velocity, acceleration, and the laws of motion have been introduced.[1] Although we do not regard this "one-week stand" as ultimately satisfactory, it allows a beginning. The method is similar to those outlined above, except that the student is already familiar with the kinematical concepts and the laws of motion. Again, programming details of the language (versions[2] exist in APL, PL/1, JOSS, FORTRAN and BASIC) are intermixed with physics. An extensive set of teacher's notes explains the material and deals with the computer situations that the teacher may encounter.

Differential Equations

Many beginning courses define velocity and acceleration using derivatives. So it is easy to write Newton's Second Law in a form involving derivatives. Then if derivatives are approximated as ratios of differences, the Euler approximation results and computer methods can be used. This approach is perhaps the oldest of those mentioned here, and in some ways the one closest to conventional elementary treatments.[3,4,5]

THE COURSE—A GEOMETRICAL APPROACH

The computer-based mechanics course, Physics 5, entailed a different approach, using the computer more extensively than those outlined above. The course was the mechanics quarter of the five-quarter introductory course for science and engineering majors. Entering students had one quarter of calculus, and the course made increasing demands on the student's knowledge of calculus. Of about 160 students in last year's class, the average student worked 12 hours at a computer terminal during the 10 weeks of the course; the time included a problem on the take-home midquarter exam which demanded computer solution. Since the course follows a non-standard approach, I prepared notes, *Notions About Motions*,[6] the organization of which is outlined in Figure 1.

We began using the laws of motion in a geometrical sense, following Newton's tactics in the first few Propositions in the *Principia*. After stating the laws of motion, Newton used them initially by considering what happens to a body under the action of impulsive forces like blows. Continuous forces are approximated by a sequence of equally timed blows, with an ever shorter time interval between weaker and weaker blows. These blows are the basis for Newton's proof of Proposition 1, the Law of Areas.

The strategy of impulsive forces allows students to very quickly get powerful results from the laws of motion, and it gives a geometrical feeling for what happens when a particle moves, corresponding closely to the computer programs introduced later. A film, *Newton's Equal Areas*,[7] displays Newton's proof of Proposition 1 and gives several examples of such motions.

The geometrical method with blows becomes analytic through imposing coordinate systems. The blow approximation leads, with no reference to

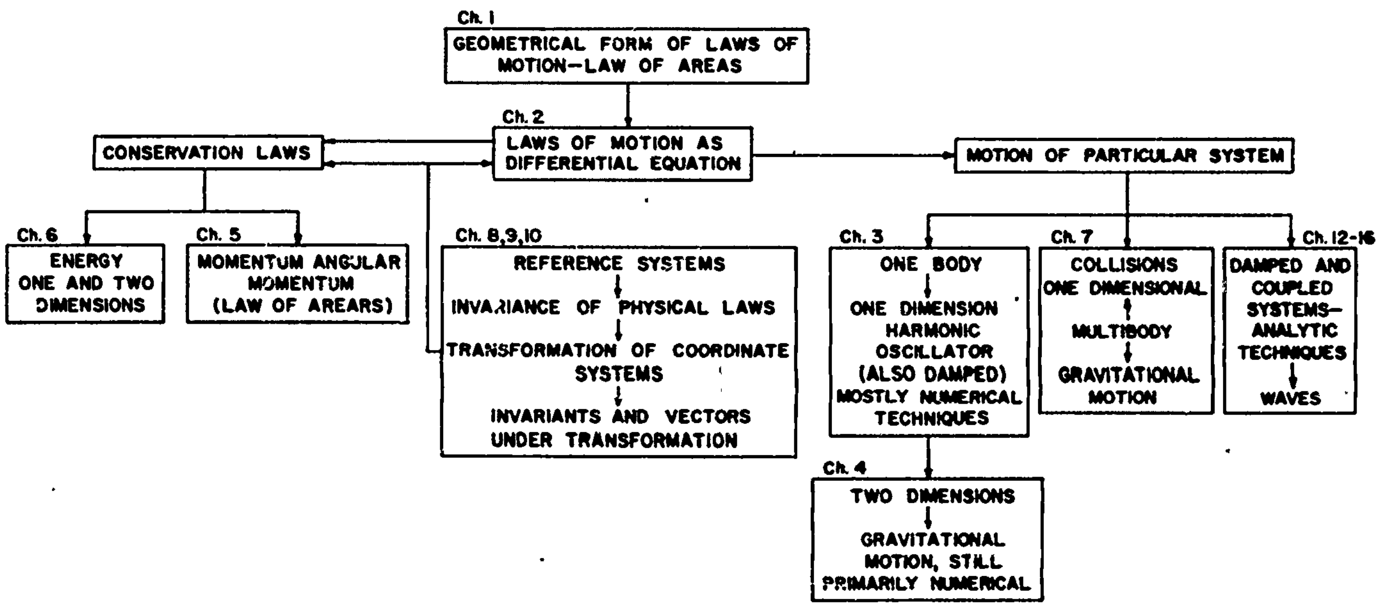


FIGURE 1. Outline of *Notions About Motions.*

differential equations, to the Euler approximations of the Newtonian equations of motion, where derivatives are replaced by ratios of differences. The transition to differential equations corresponds to the limit of more and more blows at smaller time intervals.

The Euler equations (finite time between blows) form the basis for the computer study of mechanical problems. The first physical system is the harmonic oscillator presented in five different computer languages; more difficult problems follow, first in one dimension and then in two and three; finally, with multiparticle systems. One optional problem concerns the three-body gravitational system; several students did detailed computer studies of such a system. Through numerical techniques the student is able to find how *anything* moves, whether the information about forces is in terms of analytic expressions or tables giving force as a function of position.

Physics 5 spent zero class time on details of computer programming or language. Many students had taken an introductory computer course during the first quarter. The students who had no computer experience received special assistance in the early weeks to acquaint them with the Irvine system (based on an XDS SIGMA 7) and with one or more programming languages. The computational programming languages currently available at Irvine are BASIC and FORTRAN. Neither is ideal for use with freshman science-engineering majors; I would prefer to use PL/1 or APL, regarding FORTRAN and BASIC as older, less effective languages for teaching today's students.[8] The differences in time for learning a beginning subset of any terminal-available language are not great, and we had few problems even with the students who had never seen computers before. The students are increasingly on their own in writing programs as the quarter develops.

A typical weekly assignment contained five problems plus optional problems; many problems do not involve computer usage (see Appendix A). The problems using the computer do *not* specify a programming language or terminals (some students prefer batch). Rather a problem states a physics task, and the details of how the computer is handled are left to the students. We emphasize the importance of checking a calculation, the necessity of knowing whether the program is doing what it is supposed to be doing. Thus computer programming pushes students into more hand calculation than they would normally do, in the process of showing the validity of their work.

After establishing the basic methods, we shift from numerical procedures for solving the laws of motion to analytic methods, emphasizing that for many physical problems a choice of methods exists, with sometimes one and sometimes the other, and sometimes a combination, superior. The numerical approach to differential equations *motivates* analytic approaches; when the students solve the harmonic oscillator with the computer and plot the data they usually recognize the sine function. They have recently learned to differentiate sines and cosines in calculus, so they can readily see that such a function is indeed a solution to the equation.

The course now becomes more analytic, with derivations of conservation laws using calcalus and, in the next quarter, coupled systems and wave motion using analytic techniques. Calculus is heavily used not only in lecture but also in problems; some ideas (partial derivatives for defining potential energy) are introduced before the student has encountered them in math courses.

I would not claim that freshmen became experts in using either the numerical or analytical techniques for solving differential equations. With numerical methods I avoid any discussion of methods other than the simplest Euler methods. With analytic processes the emphasis is on such procedures as guessing at solutions. But the students see and use both, and they learn of the importance of differential equations in physics.

The last section, concerned with invariance of the laws of motion under transformations, used the computer little, except for optional matrix-oriented problems on orthogonal transformations. This material, pointing toward relativity, offered little opportunity for problem-oriented computer use. In rethinking physics courses with the computer in mind, it is well to ask where its use is relevant; I would *not* claim that everything in physics lends itself to this approach, but only that the teaching of *certain* areas can be improved. Becoming too devoted to any one teaching approach is dangerous.

Although Physics 5 was mostly in the computational mode, we did have one hour of conversational mode student-machine dialogue.[9] This concerned energy conservation for a one-dimensional mechanical system. The dialogue uses calculus, and remedial sequences reflect the fact that calculus is a new tool for many students. A sample student output is shown in Appendix B.

Our experience was favorable; most students felt the dialogue was an effective learning tool. The average student spent 57 minutes with the dialogue. With feedback from the class, the dialogue has been rewritten.

This year we will have much more such dialogue material available for the course. In addition to dialogues of the proof type, we also have dialogues that assist students having difficulty solving homework problems and dialogues that simulate physical systems and allow the student to interact with them.

EVALUATION AND CONCLUSION

The goals of the beginning physics course, both for science and non-science students, are long range. Tests at the end of the year can only give a partial indication of success. One would like to follow the students in an experimental course over many years to see how these students differ statistically from other students. How do physics majors perform in junior, senior, and graduate courses? Which students becomes successful research scientists or engineers? How are attitudes about science affected by the course, now and in the future? Some longitudinal studies have been made, but they are beyond the resources of most of us.

I do not have any easy answers to the problems of evaluation, but I feel that we do need to concern ourselves with them. Student opinion in a large class such as this varies widely. Physics 5 received both very favorable and very unfavorable reviews in student evaluation. Since this way of approaching mechanics is easy for most students, it appears to some as if we are moving slowly. Although we were presenting quite advanced material compared to the usual beginning course, several good students criticized us on this score. As a result we proceeded faster and gave more difficult optional problems in the second year. Student opinion also favored more dialogue material; in 1970-71 the course will have about ten hours of dialogues available for the first two semesters.

I am extremely enthusiastic about this method of teaching beginning mechanics, in both college and high school. I would never want to teach a beginning course again where I did *not* use the laws of motion directly as differential equations. I think that the heightened level of sophistication is sufficiently great to make the "other" course unappealing. I could not, in all honesty, revert to teaching a beginning physics course which used the algebraic approaches to the laws of motion, with the typical beginning problems that one can handle that way–blocks sliding down inclined planes, projectiles, masses over pulleys, etc. Instead I believe we are entering a period of transition where more and more beginning physics courses will use the computer, and will immediately manipulate differential equations. I expect a gradual decay of the courses that teach mechanics with algebraic laws of motion, as more and more people understand the power of getting quickly to differential equations with freshman physics students.*

APPENDIX A: SAMPLE COMPUTER PROBLEMS

Problem: Using whatever computer facilities are available to you, run the harmonic oscillator calculation for several values of Δt. (Suggested range: 0.1 to 0.001.) Plot the results (position vs. time) for a representative case. If possible comment on the effect of changing Δt.

Problem: On examining your graph, can you conjecture how the position-time and velocity-time relations might be represented analytically? Check

*Teachers interested in receiving continuing information about the Physics Computer Development Project can write to the Physics Department, University of California, Irvine, California 92664.

such conjectured analytic relations, using the equations of motion as differential equations.

Problem: When a falling body is near the earth's surface, the force on it can be approximated by

$$mg-cv^2$$

where g is about 32 ft/sec^2. Study the motion for various choices of the constant c. What physical effect is represented in the second term?

Problem: A one-dimensional force of unknown origin is studied empirically. Measurements of the force are made from $x = -5$ cm to $x = 5$ cm, at intervals of 0.1 cm. Write a computer program that will use these values to find how a particle will move under the action of this force. Simulate a set of measurements, with known results, to test your program.

Problem: An anharmonic oscillator has a force law of the form

$$F = -kx - bx^3$$

Investigate the motion resulting from this force if the second term in the force is small compared to the first.

Problem: Using the computer, investigate how changing the initial condition affects the motion of a particle acted on by gravitational force. Consider carefully what strategy you will use in picking your sets of initial conditions, so that you will be able to evaluate your results. Try to find what general types of motion are possible.

Problem: Study the motion of a particle in two dimensions under the action of a force that varies directly as the distance. Use either computer or analytic methods. Is the attractive case related to a previous problem?

Problem: A small additional force acting on a particle is called a perturbing force. Suppose that the central force acting on a particle is composed of an inverse square force and a perturbing force varying inversely as the cube of the distance. At a unit distance the perturbing force is one tenth of the inverse square force. Compare the resulting motion to the non-perturbed motion.

Problem: How can energy conservation be used to check the accuracy of the calculation for the harmonic oscillator?

Problem: Write a computer program for the gravitational problem which not only prints out position but also gives the total energy each time. Remember that your calculation is an approximation; to what extent is energy conserved? Try a variety of initial conditions when you run the program.

Problem: A potential energy sometimes assumed as an approximation for the nuclear interaction is of the form

$$(A/r)e^{-r/r_0}$$

where A and r_0 are constants.

Find the force corresponding to this potential, and compare the situation to that existing for the gravitational case.

Problem: Write and run a computer program for the one-dimensional collision. Model the situation by assuming zero force until collision distance C is reached, and then assume a force which increases rapidly as the bodies become "closer." Your program might print position, total momentum, and total kinetic energy. Compare your results with those of classmates who used different collision forces.

Problem: Design a computer program for the three-body problem. Use it to explore the possible motions of a planet in a binary star system, with two suns of approximately equal mass.

Problem: Four particles move under the action of gravitational forces. (a) How many position coordinates (i.e., degrees of freedom) are needed to describe the system completely? (b) How many second-order differential equations can be written down for the system if one uses the Laws of Motion? (c) How many initial values are required to specify uniquely a particular motion?

Problem: Write a computer program that accepts a 3 x 3 matrix and determines if this matrix represents an orthogonal transformation.

APPENDIX B: SAMPLE DIALOGUE

```
WHAT DO YOU WANT TO MULTIPLY THE
EQUATIONS OF MOTION BY?
SOME RELEVANT QUANTITIES ARE
        M = MASS
        X = POSITION
        V = VELOCITY
        A = ACCELERATION
        F = FORCE

?LET'S TRY THE MASS

SINCE M HAS ZERO DERIVATIVES, IT DOES NOT
CONTRIBUTE TO FINDING AN EXPRESSION OF
THE FORM D/DT(SOMETHING)
TRY AGAIN.

?HOW ABOUT THE VELOCITY?
FINE . . .
THE RESULT IS M*V*DV/DT=F*V

NOW YOU ARE IN A POSITION TO TRANSFORM
THE EQUATION TO THE FORM
```

D/DT (SOMETHING) = O.

DO YOU WANT TO BEGIN BY TRANSFORMING THE RIGHT SIDE, OR THE LEFT SIDE?

?I WANT TO TRY THE RIGHT HAND SIDE. PLEASE

FIRST, LET'S TALK A LITTLE ABOUT FORCES. WHAT PHYSICAL VARIABLES DO FORCES DEPEND ON?

?POSITION. . . ?

THE VARIABLES OF INTEREST TO US ARE THE TIME DEPENDENT VARIABLES.

POSITION. X(T)
VELOCITY. V(T) AND THE TIME. T. ITSELF.

HERE ARE THREE EXAMPLES OF ONE-DIMENSIONAL SYSTEMS SUBJECT TO FORCE LAWS YOU MAY HAVE SEEN:

HARMONIC OSCILLATOR: F = – K*X
DAMPED OSCILLATOR: F = –K*X –A*V
INVERSE SQUARE FORCE: F = – C/X↑2

THESE FORCES DO NOT DEPEND ON TIME EXPLICITLY BUT THE SYSTEMS WHOSE MOTIONS THEY INFLUENCE. EXPERIENCE CHANGING FORCES BECAUSE THEIR POSITIONS AND VELOCITIES CHANGE WITH TIME.

LET'S TRY THE SIMPLEST CASE, WHERE THE FORCE DEPENDS ONLY ON POSITION. CAN YOU GIVE ME THE "SOMETHING" WHOSE TIME DERIVATIVE IS THE RIGHT HAND SIDE OF OUR EQUATION.

F*V

THE RESPONSE I AM LOOKING FOR IS

F*DG/DT+DF/DT*G.

IF YOU NEED ASSISTANCE WITH THE DETAILS OF THE CALCULATION, YOU MIGHT SEE THOMAS. CALCULUS. PAGES 71 AND 72.
OR JOHNSON/KIOKEMEISTER, CALCULUS, PAGES 117-119.

A SPECIAL CASE IS F = G. WHAT IS THE TIME DERIVATIVE OF

F↑2?

?2F DF/DT

GOOD....
DOES THIS PROPERTY OF F↑2 SUGGEST THE SOMETHING SUCH THAT

D/DT (SOMETHING) = M*V*DV/DT?

?PERHAPS

SECOND HINT: TRY TAKING THE DERIVATIVE WITH RESPECT TO TIME OF

C*F↑2

WHERE C IS A CONSTANT.

?2CFDF/DT

YOUR DERIVATIVE IS CORRECT.
D/DT (C*F ↑ 2)=2*C*F*DF/DT, WHEREAS
D/DT (SOMETHING)=M*V*DV/DT.

WHAT IDENTIFICATIONS OF F AND C MAKE THE RIGHT SIDES OF THE ABOVE EQUATIONS THE SAME?

?F=V*V.C=M/2

THEN WHAT IS THE "SOMETHING" SUCH THAT
D/DT (SOMETHING) = M*V*DV/DT ?

?M V ↑2

SUBSTITUTING YOUR IDENTIFICATIONS FOR F AND C INTO THE LEFT HAND SIDE OF THE EQUATION GIVES: SOMETHING =(1/2)*M*V↑2.

YOU MIGHT CHECK BY DIFFERENTIATION THAT

D/DT((1/2)*M*V↑2)=M*V*DV/DT

YOU MAY RECOGNIZE THE EXPRESSION

(1/2)*M*V↑2

IF SO, ENTER ITS NAME.

?THE ENERGY

IT'S CALLED THE KINETIC ENERGY.

REMEMBER THAT WE HAVE BEEN WORKING WITH THE NEWTONIAN EQUATIONS OF MOTION. IF WE HAD STARTED WITH MORE GENERAL EQUATIONS OF MOTION, SUCH AS THOSE OF RELATIVISTIC MECHANICS, WE WOULD OBTAIN A DIFFERENT QUANTITY THAT WE MIGHT WANT TO CALL "KINETIC ENERGY."

NOW LET'S TRY TO TRANSFORM THE OTHER SIDE.

REFERENCES

1. R. Blum and A. M. Bork, "Computers in the Physics Curriculum," *Am. J. Phys.*, *38*, 959 (1970), gives a more detailed discussion.
   A. M. Bork, "Computers in High School Physics," *The Physics Teacher*, September 1969.
2. A. M. Bork, A. W. Luehrmann, and J. W. Robson, in *Introductory Computer-Based Mechanics*, R. Blum (ed.), Commission on College Physics, College Park, 1969.
3. C. Sherwin, *Basic Concepts of Physics*, Holt, Reinhardt and Winston, New York, 1961.
4. R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison-Wesley, Reading, Mass., 1961.
5. A. M. Bork, *Fortran for Physics*, Addison-Wesley, Reading, Mass., 1966.
6. A. M. Bork, *Notions about Motions*, W. H. Freeman, San Francisco, 1971.
7. B. Cornwell, C. Comwell, and A. M. Bork, *Newton's Equal Areas*, International Film Bureau, Chicago, 1967.
8. A. M. Bork, *Physics Teaching and Computer Languages*, preprint, Dept. of Physics, University of California, Irvin, 1969.
9. A. M. Bork and N. Sherman, "A Computer-Based Dialogue for Deriving Energy Conservation for Motion in One Dimension,' *Am. J. Phys.*, 1971 (in press). A flow-chart of this dialogue is also available.

A Strobe-Computer Laboratory for Introductory Physics

ELISHA R. HUGGINS

INTRODUCTION

When a strobe photograph is used in the experimental definition of velocity and acceleration, one is analyzing the motion of a particle at short but discrete time intervals in precisely the same way that motion is described in many computer programs. With the availability of a time-shared computer, a student can write a simple program based on the strobe definitions and use it to analyze experiments in the laboratory. Having developed confidence in both his program and the use of the computer, he can then go beyond elementary lab projects and use the computer to predict the motion of spacecraft, the effects of changing Newton's law of gravity, the motion of electrons in electric and magnetic fields, or a number of other topics of his own choosing.

Several years ago at Dartmouth we developed a strobe laboratory for use in the introductory physics courses. For the science majors, the labs provided several interesting experiments; for liberal arts students the strobe labs along with graphical analysis and experimental definitions of velocity and acceleration replaced calculus and served as the basis of all studies in mechanics.[1]

In the last two years, we have had considerable success in combining the strobe labs with computer work to serve as a natural introduction both to mechanics and the use of the computer. Aside from the (now) expected result of generating in nearly all students a considerable interest in mechanics, the use of the computer has had a rather surprising effect on the non-science students. Previously a course for non-science majors was considered clearly "second rate," because it either did not use or did not emphasize calculus. Now, when a pre-med history major can solve a three-body problem on a computer. and a senior English major can use the Yukawa potential and compute the effect of a graviton rest mass on a satellite's orbit, no feeling of inferiority remains.

Department of Physics, Dartmouth College, Hanover, New Hampshire. Work supported by the National Science Foundation.

STROBE LABS AND GRAPHICAL ANALYSIS

We shall here briefly describe the strobe laboratory set-up and the techniques of graphical analysis of strobe photographs.[1] Our experimental set-up is shown in Figure 1. It consists of a grid with strings one centimeter apart, with every tenth string dark. The grid is permanently mounted about fifteen inches in front of a black painted wall. A strobe flash lamp is located beneath the grid so that it illuminates the ball and not the grid lines. The grid lines are illuminated by a permanently mounted spotlight whose intensity is accurately controlled by a Variac, and whose beam is aimed at the ceiling to produce an even, glare-free light. The Polaroid camera, situated over a permanently marked spot on the floor, contains the high speed (ASA 10,000) Polaroid

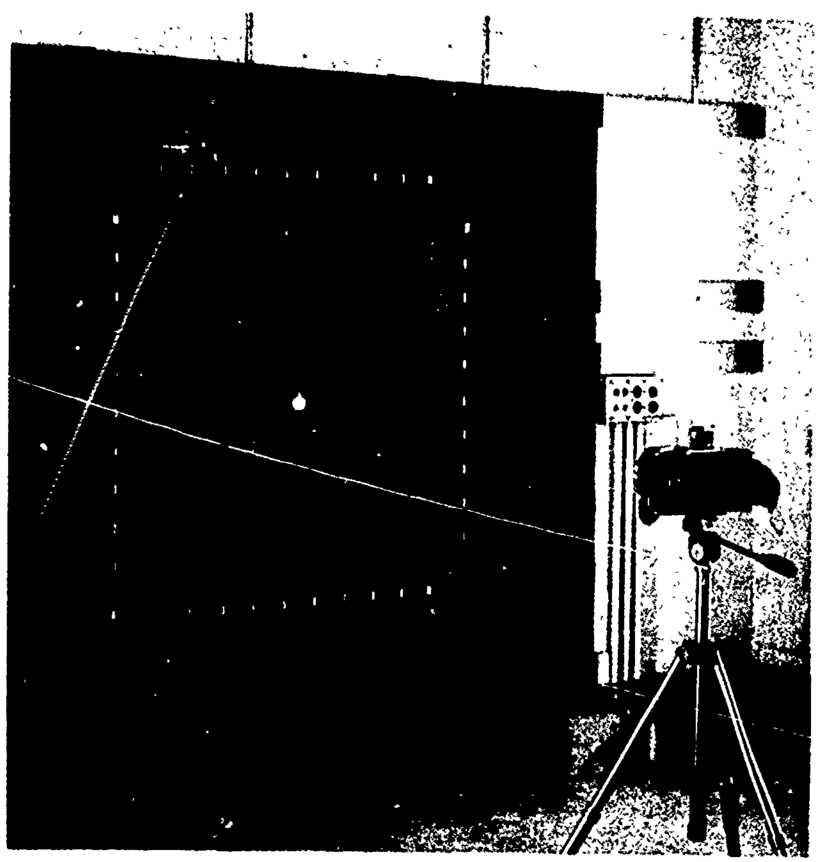


FIGURE 1 Laboratory set-up for taking strobe photographs. Set-up time is reduced by permanently marking the location of the strobe flashing unit (below the grid) and Polaroid camera.

film. The permanent mounting of the equipment has eliminated storage problems and greatly facilitated setting up of the experiments.

In Figure 2 we see a strobe photograph of projectile motion taken with a time interval $\Delta t = 0.1$ sec. Note that the grid has been labeled so that it imitates graph paper and serves as the coordinate system for all analysis. To facilitate analysis, we have constructed our own graph paper which looks like the grid, and is exactly 7 inches x 7 inches to match a teletype plot. As a result a student can directly compare his graphical analysis and computer plot by holding them together up to the light.*

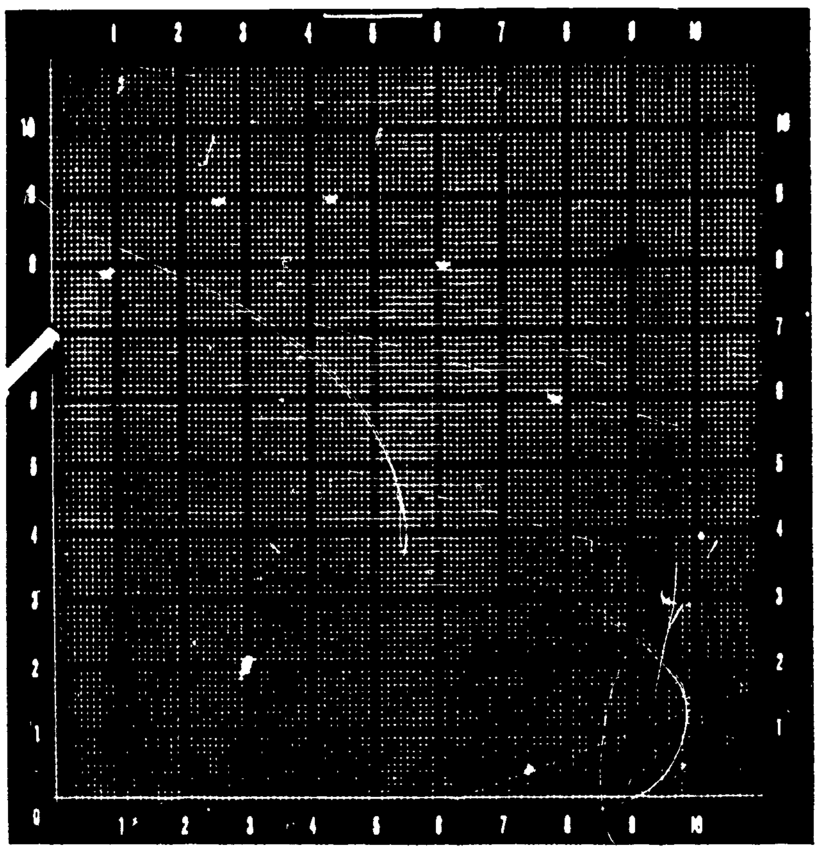


FIGURE 2 Strobe photograph of steel projectile.

*The graph paper or a metal multilith master may be obtained from Roger Burt, Printers, of Hanover, N.H. 03755.

Using the experimental definitions of velocity and acceleration shown in Figure 3, we obtain (Figure 4) a graphical analysis of the projectile motion photograph. This analysis demonstrates the main feature of projectile motion, that the particle's acceleration vector $\vec{A}$ is constant throughout the particle's $\vec{A}_1, \vec{A}_2, \vec{A}_3$ and $\vec{A}_4$ are indeed parallel, of equal length, and also point downward.

WRITING A COMPUTER PROGRAM

In Figures 5 and 6 we have rewritten the experimental definitions of $\vec{V}$ and $\vec{A}$ shown in Figure 3 so that these definitions can be used to predict the motion of the particle in a step by step manner. For example, Figure 5 illustrates how, knowing $\vec{R}_{old}$ and $\vec{V}_{old}$, we can calculate the next position

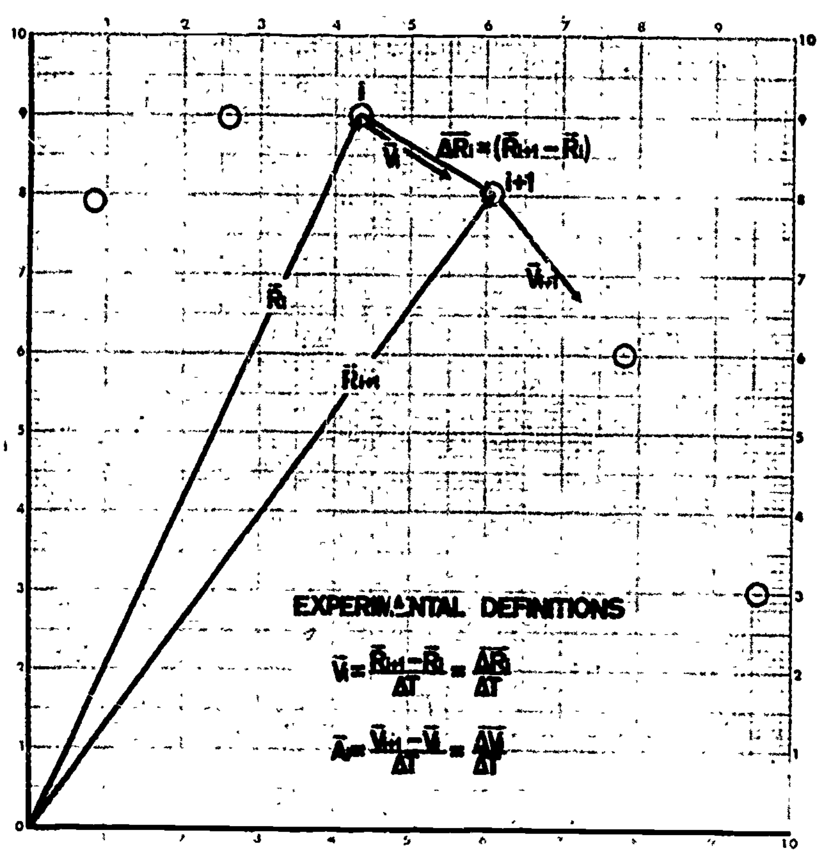


FIGURE 3 Use of a strobe photograph to provide experimental definitions of velocity and acceleration.

vector $\vec{R}_{new}$. Figure 6 shows how to determine the next velocity vector if the acceleration $\vec{A}$ is known. The next obvious step, which can however occur quite a bit later in a physics course, is to use Newton's second law $\vec{A} = \vec{F} M$ to determine $\vec{A}$ and complete the process. This graphical approach follows closely the analysis described by Bork, Luehrmann, and Robson in their excellent monograph.[2]

In order to predict the next position of an object, one must establish the initial conditions. Figure 7 shows an accurate method of obtaining the instantaneous velocity of a particle at the initial point O for use as the initial velocity. One needs the instantaneous velocity at O for a computer analysis because the computer will use a much shorter time step D than the step Δt used in the strobe photograph.

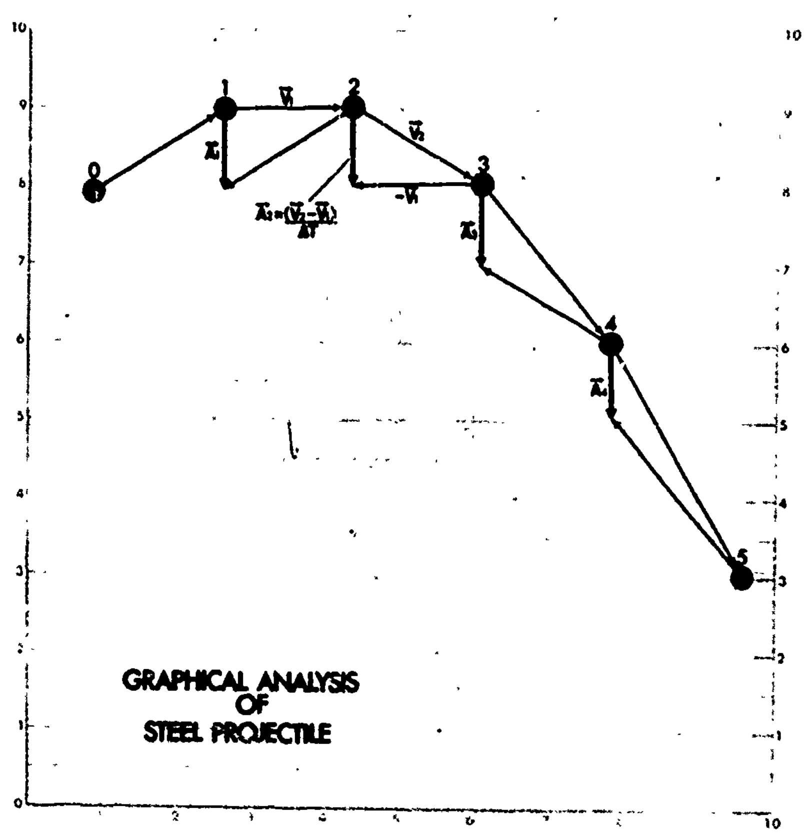


FIGURE 4 A graphical analysis of the motion of the steel projectile demonstrates that the projectile's acceleration A is constant throughout its motion.

In Figure 8, we have a computational algorithm based on the definitions of $\vec{R}_{new}$, $\vec{V}_{new}$, and $\vec{V}_0$ given in Figures 5, 6 and 7. At this point the emphasis is upon the fact that the program represents a set of *instructions* that are to be followed precisely in order to obtain a prediction of the motion of the projectile. The student should try to follow these instructions himself, then translate them into a computer language as shown, for example, in Figure 9.

COMPUTER ANALYSIS OF STROBE LAB EXPERIMENTS

In Figure 9, the algorithm of Figure 8 has been translated to the BASIC computer language, and executed. This version has been kept as simple as possible and does not have the appropriate labeling of output columns. However, even with labeling, numerical output does not have the impact or convenience of the graphical output seen in the teletype plot of Figure 10.

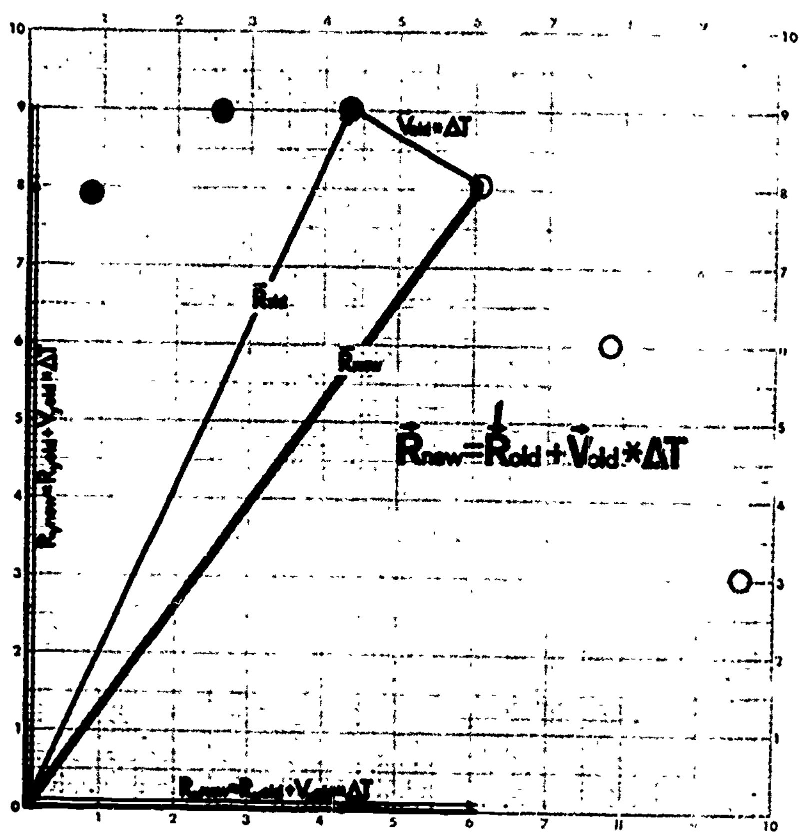


FIGURE 5 Formula for the next position vector of the particle.

To obtain such a plot, our beginning students simply merge their program with a special library program which automatically produces a graph with borders that match the grid. A more advanced student would use our general teletype plotting programs SCALE and DRAW, following a set of plotting rules that apply to both teletypes and *XY* recorders. A brief example of the use of these plotting programs will be given at the end of this paper.

Our basic approach in introducing computer work is to have the student become familiar with a simple introductory program, explicitly the one for projectile motion. Then, by a series of minor modifications in the familiar program, we work toward the solution of more sophisticated and original problems. In this way a student with no previous computer experience can obtain understandable results without an enormous amount of outside help. Our reason for adopting this approach was that in our early limited ex-

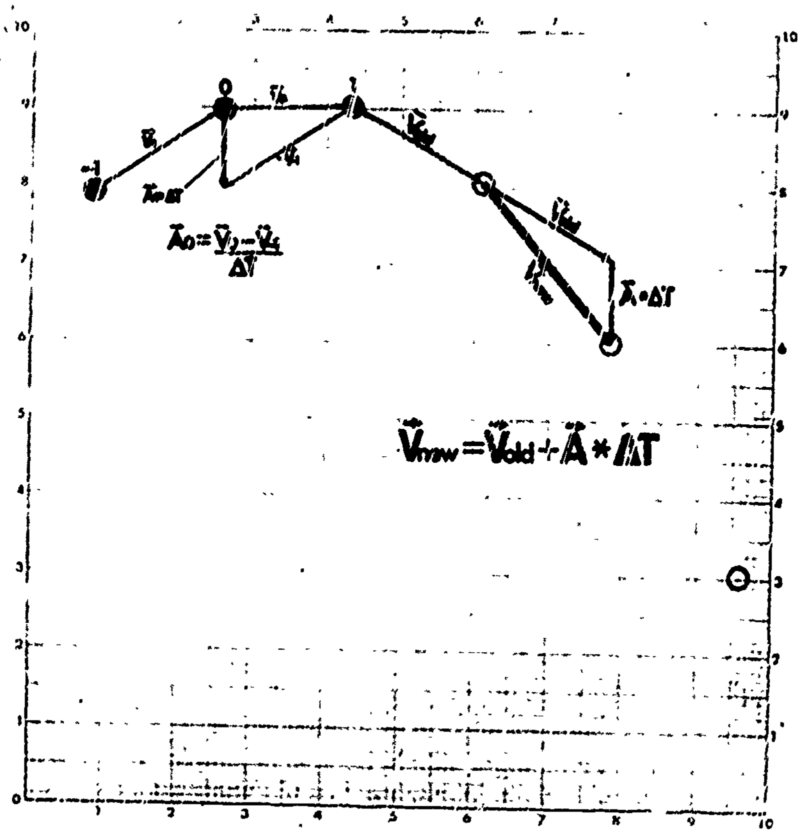


FIGURE 6 Formula for the next velocity vector of the particle.

periments with the use of the computer, inexperienced students rashly attempted formidable problems and required hours of individual attention. This is simply not feasible in large courses.

The first modifications a student will make in the projectile motion program will be to change initial conditions to match those in his experiment. He can then adjust various parameters to obtain the best possible agreement between his computer plot and his experimental results. This procedure is one of the most vivid and enjoyable forms of error analysis yet devised for an introductory lab.

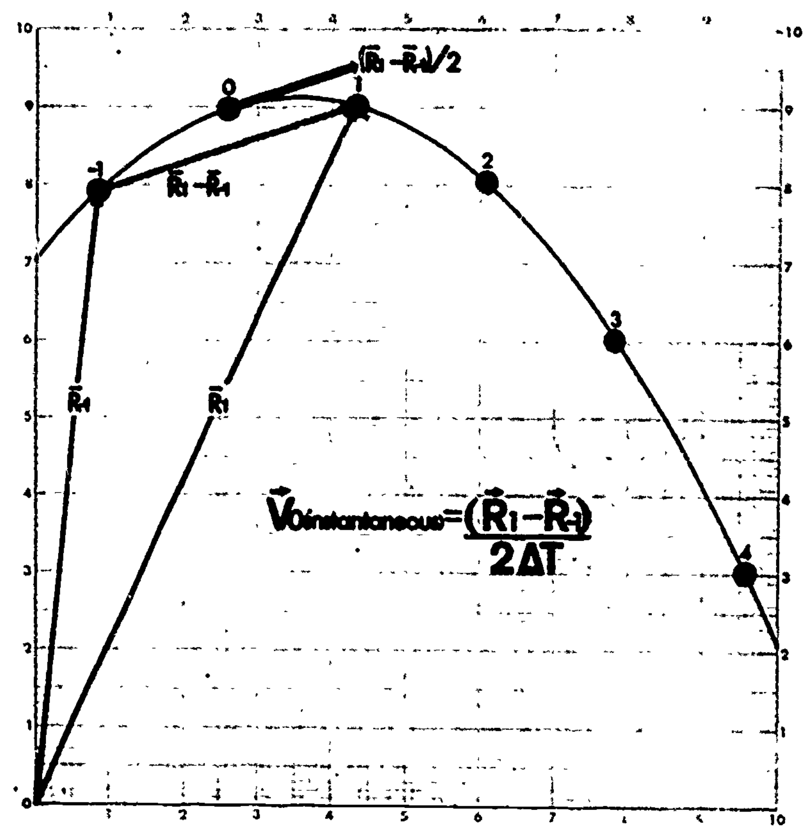


FIGURE 7 Calculation of the instantaneous velocity. For projectile motion the instantaneous velocity is equal to the average of the preceding and following strobe velocity vectors, and is determined as shown. This procedure gives quite accurate results for other kinds of strobe photographs.

```
INITIAL CONDITIONS
    LET R⃗old = R⃗o
    LET V⃗old = (R⃗1 − R⃗−1)/(2*ΔT)
    LET T = 0
TIME STEP
    LET D = .01
CALCULATIONS
    LET R⃗new = R⃗old + V⃗old*D
    LET A⃗ = g⃗
    LET V⃗new = V⃗old + A⃗*D
    LET Tnew = Told + D
PRINTING
    PRINT T, R⃗, V⃗
LOOP
    REPEAT CALCULATIONS
    FOR NEXT Tnew R⃗new V⃗new
```

FIGURE 8 An algorithm for calculating projectile motion.

```
100'---------------INITIAL CONDITIONS
110 LET R1=25.9
120 LET R2=89.9
130 LET V1=(43.2-8.3)/(2*.1)
140 LET V2=(90.2-79.3)/(2*.1)
150 LET T=0
160'---------------TIME STEP
170 LET D=.01
180'---------------CALCULATIONS
190 FOR N=1 TO 10
200 LET R1=R1+V1*D
210 LET R2=R2+V2*D
220 LET A1=0
230 LET A2=-980
240 LET V1=V1+A1*D
250 LET V2=V2+A2*D
260 LET T=T+D
270 NEXT N
280'---------------PRINTING
290 PRINT T,R1,R2,V1,V2
300'---------------LOOP
310 GO TO 190
320 END
READY

RUN

STROBE      30 SEP 70   00:12

 0.1          43.35        90.94        174.5        -43.5
 0.2          60.8         82.18        174.5        -141.5
 0.3          78.25        63.62        174.5        -239.5
 0.4          95.7         35.26        174.5        -337.5
 0.5          113.15       -2.90003     174.5        -435.5
 0.6
STOP
```

FIGURE 9 The BASIC program for the algorithm of Figure 8.

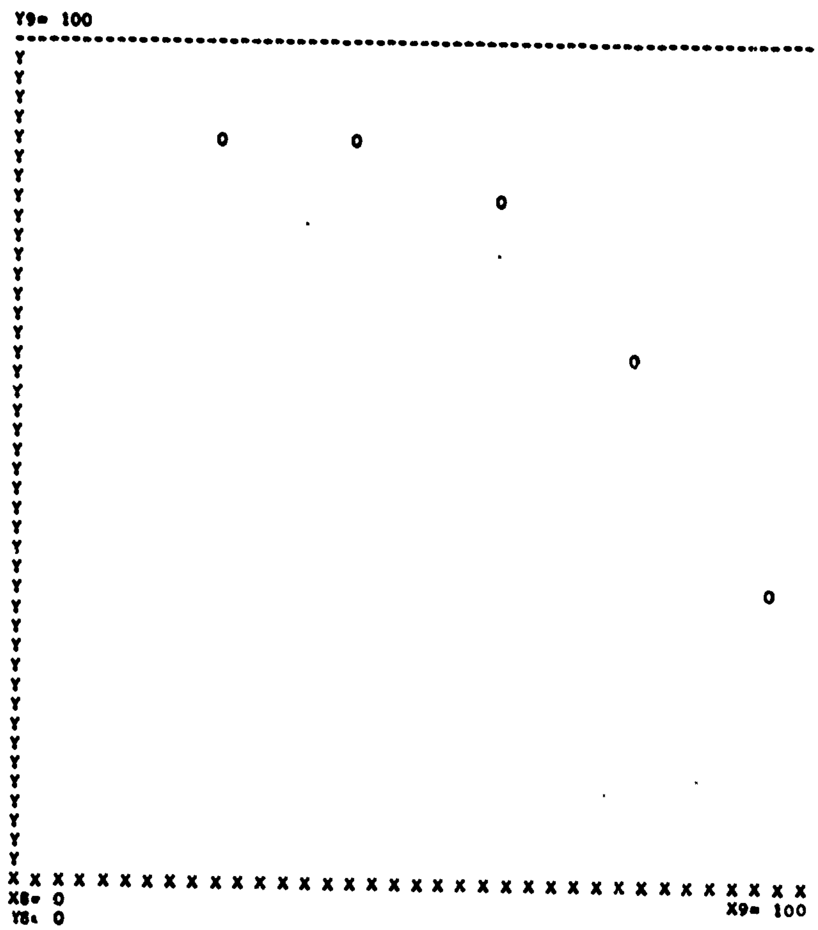


FIGURE 10 Teletype plot of projectile motion.

SIMULATION

Once the student has become familiar with the standard projectile motion problem, he studies the motion of a projectile whose acceleration is modified by the effects of air resistance. Figure 11 shows a strobe photograph of a

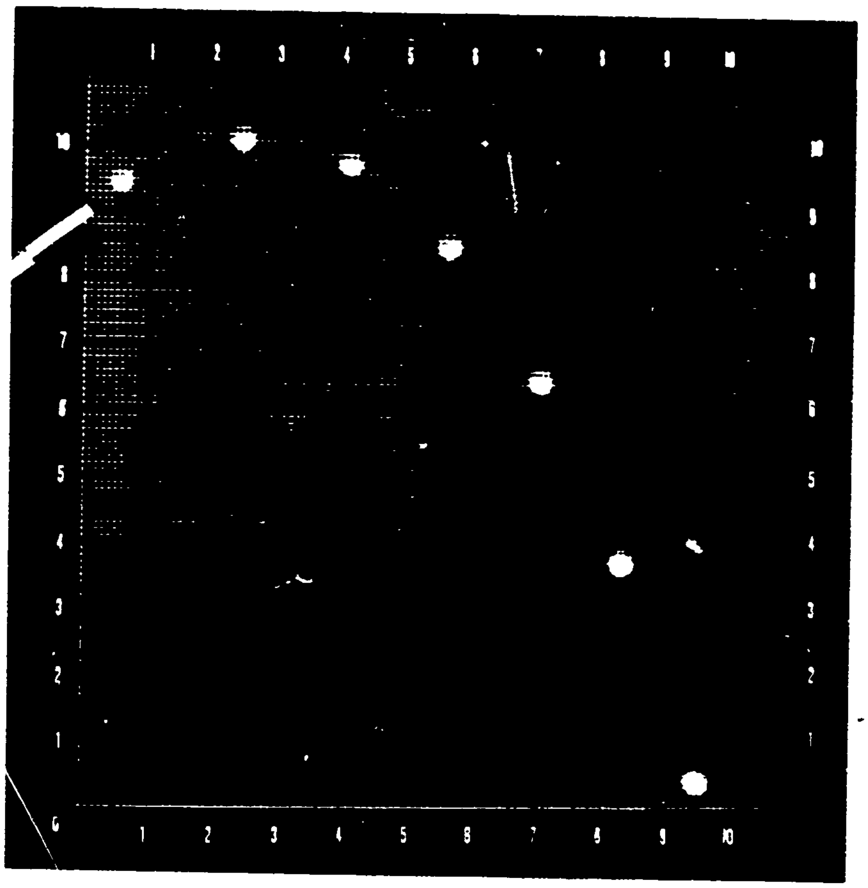


FIGURE 11 Strobe photograph of a styrofoam projectile. This is similar to Figure 2 except that the steel ball is replaced by a styrofoam ball to maximize the effects of air resistance.

styrofoam projectile; the graphical analysis shown in Figure 12 demonstrates that $\vec{A}$ is modified by a vector of the form $\vec{A}_{air} = -K\vec{V}$. To analyze this experiment on the computer, the student merely modifies commands defining $\vec{A}$ in his projectile motion program of Figure 9:

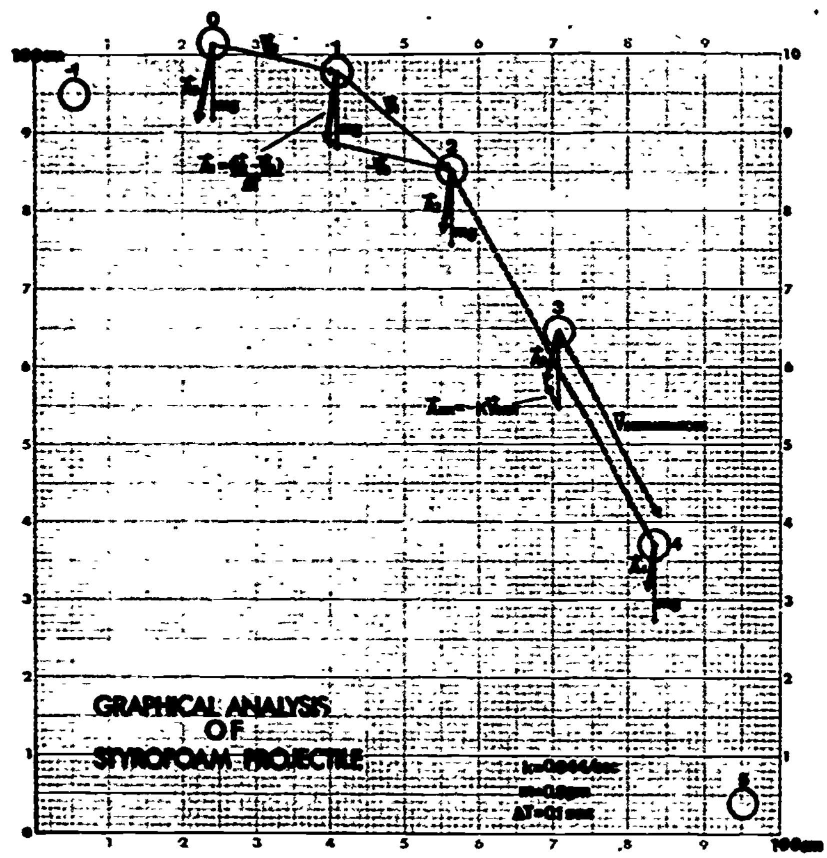


FIGURE 12 Graphical analysis of the styrofoam projectile. The acceleration vectors are deflected backwards by air resistance oppositely directed to the ball's instantaneous velocity. Even with a styrofoam ball, extremely accurate graphical work is required to obtain this result.

```
LET A1 = 0
LET A2 = –980
```

are changed to

```
LET K = ...
LET A1 = –K*V1
LET A2 = –980–K*V2
```

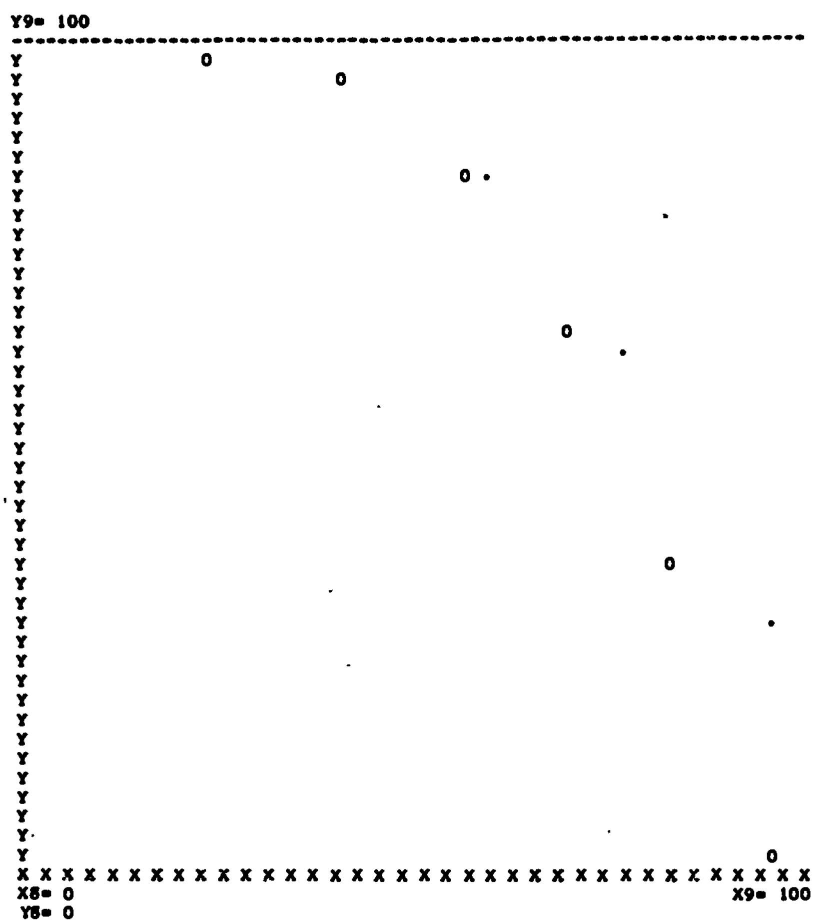


FIGURE 13 Computer analysis of the styrofoam projectile. The dots show where the ball would have gone in the absence of air resistance.

where the student determines K by trial and error until he obtains the best agreement between his experimental results and his teletype plot. In Figure 13 this agreement was obtained with the value K = .944; with the graphical work, one could merely determine that K was approximately 1. The dots represent the motion under the same initial conditions, but without air resistance.

Projectile motion with air resistance provides an excellent illustration of

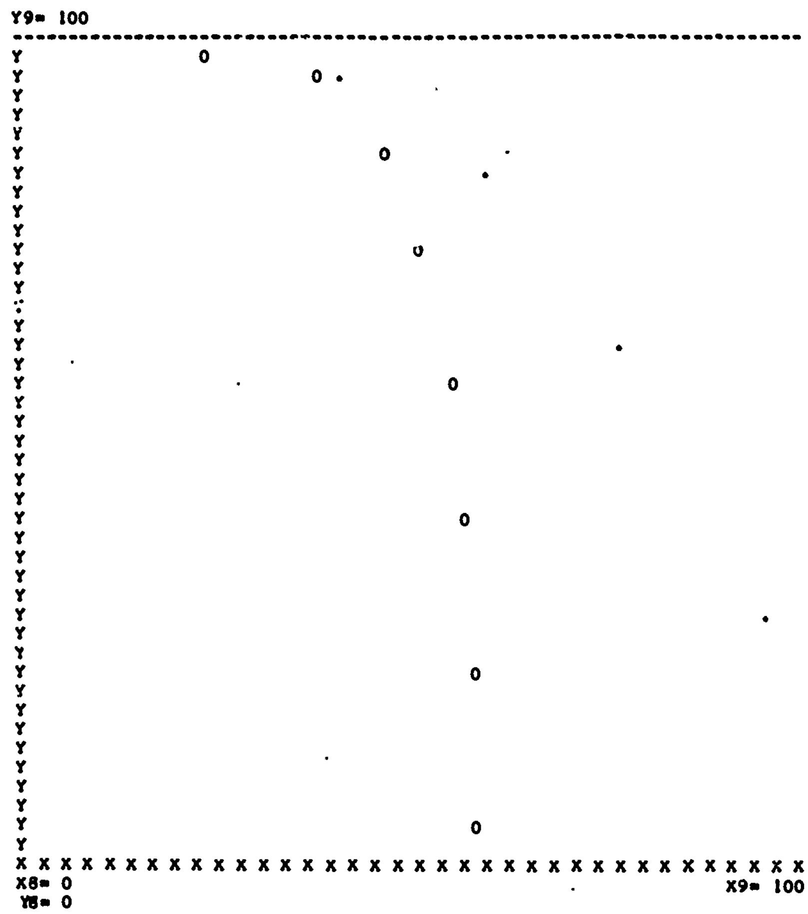


FIGURE 14. Increasing air resistance, K = 5.

computer experiments which simulate situations that would be too expensive or too difficult for experimental study in the classroom. Even in the case of the styrofoam ball, the lightest conveniently obtainable, the effect of air resistance, though appreciable, is not great on the small scale of a laboratory experiment. However, it is extremely easy to simulate a greater air resistance on the computer simply by increasing the constant K. In Figure 14, where k

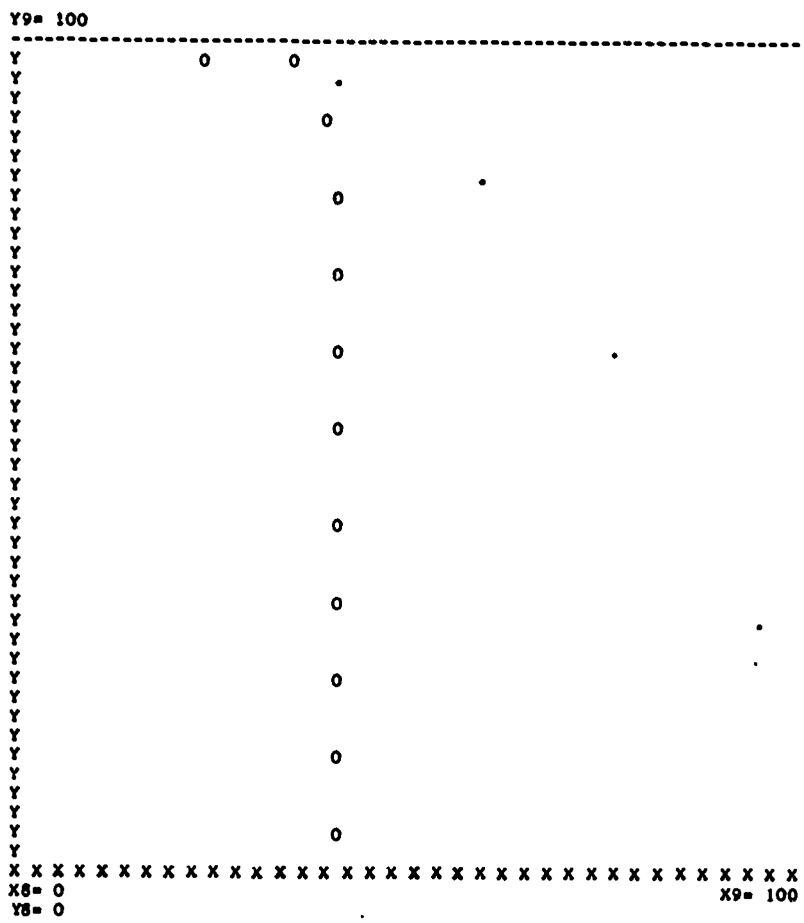


FIGURE 15 Increasing air resistance, K = 10.

is increased by a factor of 5, there is a noticeable change in the motion of the ball.

In fact a new feature begins to appear. If one is observant he notices that the ball reaches a terminal velocity about half way down. With a further increase in K from 5 to 10 (Figure 15), the terminal velocity become the main feature of the motion. As a project the student can "experimentally"

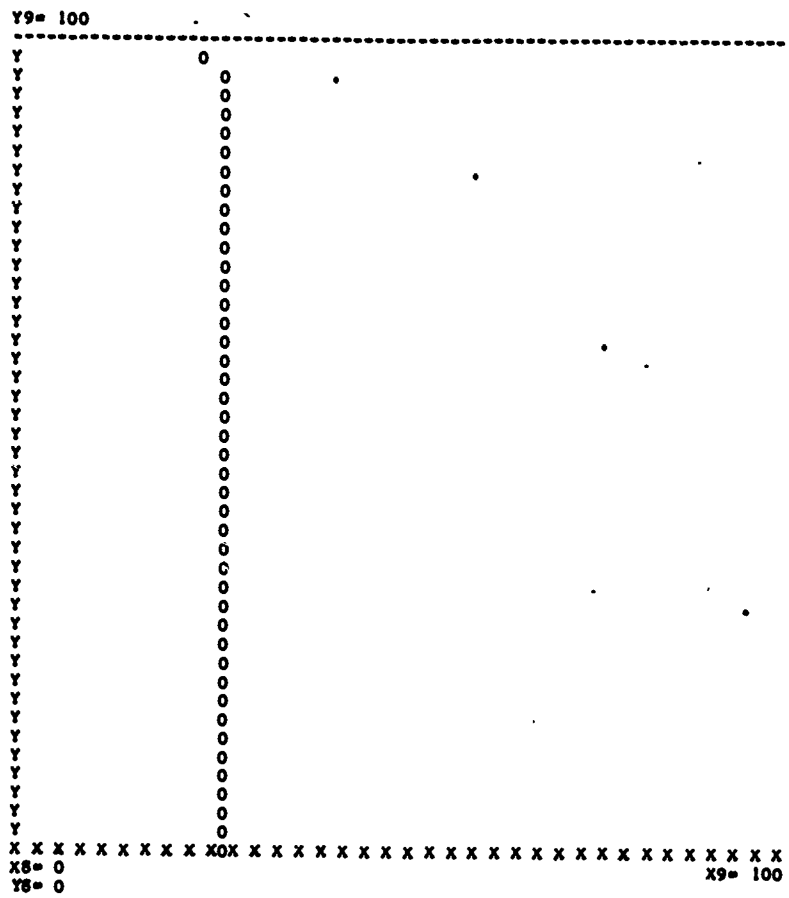


FIGURE 16 A marble falling in molasses in January.

study the relation between K and the terminal speed, or he can increase K much further as in Figure 16 to see how a marble falls through molasses.

The final strobe lab we have used involves a situation which can be analyzed either graphically or by computer, but is too complicated to be handled by standard analytical techniques. Figure 1 showed a ball hanging on the end of a spring in front of the grid. If the ball is released in front of the grid at a

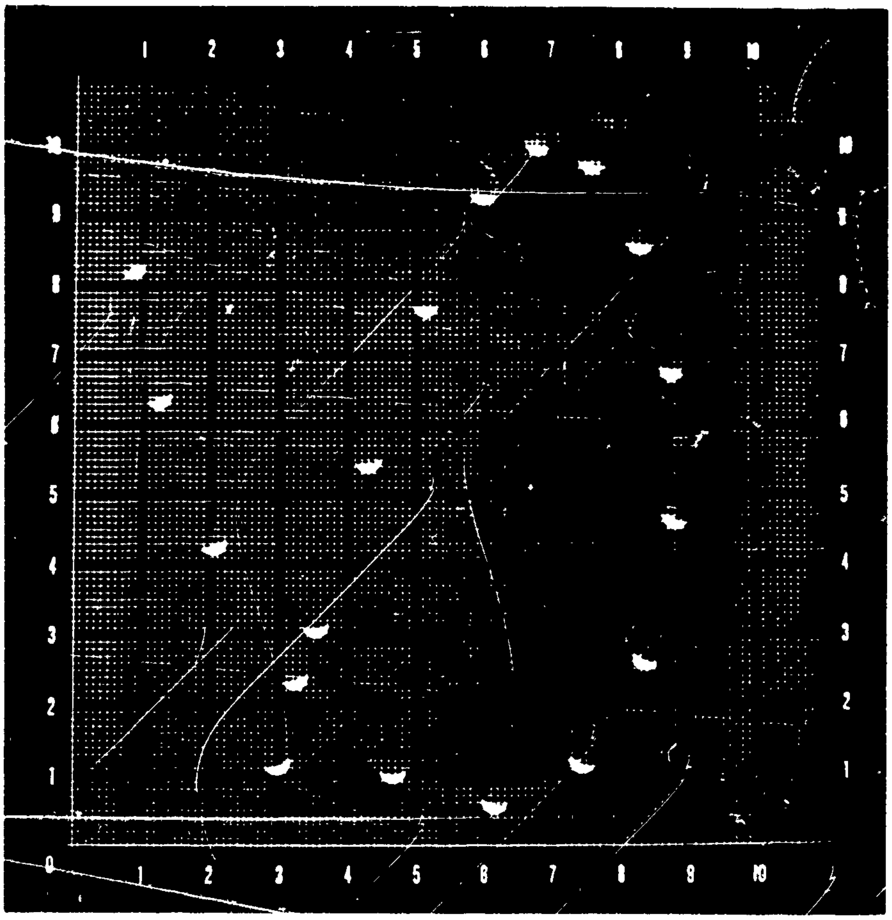


FIGURE 17 Ball bouncing on a spring. Figure 1 shows the ball resting on the end of its spring. For this photograph, taken with Δt = .1 sec., the ball was released at coordinates (90, 90) and allowed to bounce in front of the grid during a time exposure.

point approximately given by the coordinates (90,90), one can obtain a strobe photograph similar to that shown in Figure 17. The simplest approach is to modify A1 and A2 in the program of Figure 9 to include the horizontal and vertical components of the Hooke's Law restoring force in the spring, which is always directed back toward the nail which holds the spring to the wall. The strobe photograph of Figure 17 compares very well to the result of

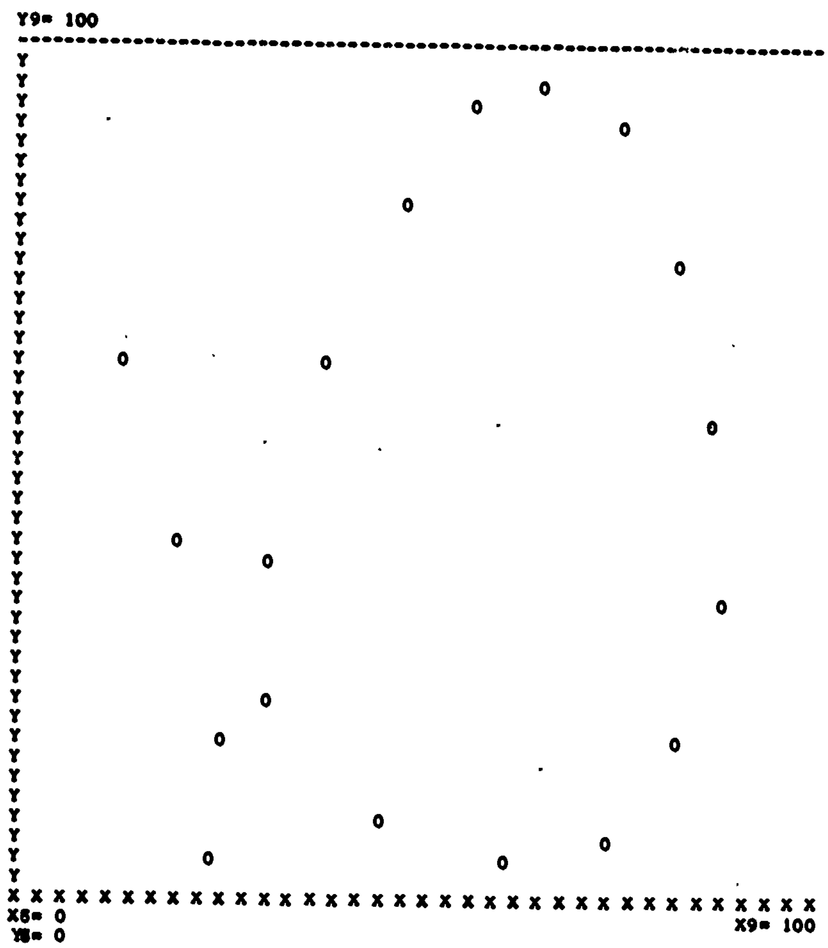


FIGURE 18 Teletype plot of the ball bouncing on a spring. If the student has transferred the results of his strobe photograph to our 7 x 7 inch graph paper, he can compare his experimental results with the computer plot simply by holding them together up to the light.

the computation, Figure 18, demonstrating that the computer can handle this exercise rather well, considering the fact that sixteen consecutive points were predicted using only the first three to determine initial conditions.

TELETYPE PLOTTING [3]

The use of the teletype plotting programs SCALE and DRAW is shown in Figure 19, where we have modified the projectile motion program of Figure 9 to obtain the teletype plot shown in Figure 10. Here we have followed a set

```
10 SUB SCALE;DRAW
20 LET X8=Y8=0
30 LET X9=Y9=100
40 GOSUB#1
100'---------------INITIAL CONDITIONS
110 LET R1=25.9
120 LET R2=89.9
125 LET X1=R1
126 LET Y1=R2
127 LET P1=2
128 GOSUB#2
130 LET V1=(43.2-8.3)/(2*.1)
140 LET V2=(90.2-79.3)/(2*.1)
150 LET T=0
160'---------------TIME STEP
170 LET D=.01
180'---------------CALCULATIONS
190 FOR N=1 TO 10
200 LET R1=R1+V1*D
210 LET R2=R2+V2*D
220 LET A1=0
230 LET A2=-980
240 LET V1=V1+A1*D
250 LET V2=V2+A2*D
260 LET T=T+D
270 NEXT N
280'---------------PRINTING
290 LET X1=R1
291 LET Y1=R2
292 LET P1=2
293 GOSUB#2
300'---------------LOOP
310 IF R2>=0 THEN 190
311 LET P1=0
312 GOSUB#2
320 END
```

FIGURE 19 Projectile motion program of Figure 9 modified so that it will produce the teletype plot shown in Figure 10.

of rules which apply to both teletype and *XY* recorder plotting, the only difference being the name of the subprograms called.

In shortened version, the rules are the following:

1) Name the subprograms to be used. (Line 10 of Figure 19. This is a requirement of the Dartmouth system as of Sept. 30, 1970.)

2) Let (X8,Y8) be the coordinates of the lower left hand corner and (X9,Y9) be the coordinates of the upper right hand corner of the desired plot. Enter these values of X8, Y8, X9, Y9 and call SCALE by means of the "GOSUB #1" command. (Lines 20, 30, and 40 of Figure 19. This sets the scales for all plotting.)

3) When you wish to plot a point, set
X1=X coordinate of point to be plotted
Y1=Y coordinate of point to be plotted
P1= number of desired plotting character
and then call DRAW ("GOSUB #2") (This is done in lines 125-128 to plot the initial point, and in lines 290-293 to plot the calculated points. In line 290 we wiped out the old printing command so that numerical tables would not appear inside the plot. Setting P1=2 has the points plotted in the second plotting character. namely zero. P1=1 produces an, P1 = 3 a period, etc.)

4) The program DRAW stores points in a file until a final printing command is encountered. In this way multi-valued functions can be plotted even though teletypes cannot backspace. The final print command is to set P1= 0 and go back to DRAW, (as seen in lines 311 and 312).

If one wants a line drawing plot produced by an *XY* recorder plotter, he merely changes the name of the sub programs in line 10. Thus teletype plotting can be used to debug programs for line drawings when *XY* recorder plotters are not immediately available. The idea of making teletype plot commands compatible with *XY* recorder commands, and the initial work on the teletype plotting programs was that of a Dartmouth student Gus Zimmerman.

In some cases, there may be such a profusion of data points that the teletype plot becomes illegible. This can be avoided, in part, by a special command which instructs DRAW to change to a new character for plotting. If, however, there are still too many points for legibility, DRAW can be instructed to advance to a blank page and start a new graph from the place at which the previous graph reached its specified number of points.

REFERENCES

1. E.R. Huggins, *Physics 1*, W.A. Benjamin, Inc., New York, 1968, gives a more complete description of setting up a strobe lab and of experimental definitions of velocity and acceleration.
2. A. Bork, A. Luehrmann, and J. Robson, *Introductory Computer-Based Mechanics, A One-Week Sample Course*, R. Blum (ed.). This monograph may be obtained on request from the Commission on College Physics, University of Maryland, College Park, Maryland.
3. J. R. Merrill, "Inexpensive, Interactive Graphics on Telephone-Connected, Time-Shared Computers," in this volume, offers more information on this subject.

The Use of Computers in Teaching Calculus

HARRY M. SCHEY

The Education Research Center of MIT at present is undertaking an experiment which is a great departure from conventional learning patterns. This is the Unified Science Study Program (USSP); a project-oriented, self-paced and multidisciplinary program run last year with about fifty freshmen from MIT and Tufts University; this coming year there will be about one hundred students enrolled in the program, most of them from MIT, but some from several surrounding community colleges, and the program will be extended to include sophomores as well as freshmen. Students working in this program engage in a project or series of projects and learn in response to the need for knowledge arising from the projects. The program is self-paced, breaking the lock-step lecture-laboratory-recitation regime of the traditional college program, and in its multidisciplinary character it deliberately blurs the boundaries that commonly separate traditional disciplines. Nature itself is not compartmentalized, and so the way we learn about nature does not require organization of knowledge into departmental disciplines.

The mathematics part of the program, which is the work of Jeffrey Nicoll, Judah Schwartz, William Walton, Jerrold Zacharias and myself, is called a *Laboratory, Computer and Calculus Based Course in Mathematics.* As the title implies, the course has some rather unique aspects. Not the least of these is its use of a laboratory, the inclusion of which is motivated by a number of considerations. To begin with, the course, as it is currently structured, is intended to serve students who are engaged in learning, totally or in part, through laboratory work. Perhaps more importantly, the laboratory will stimulate in the student that most important of all "prerequisites," the need and desire to know. These arise in a student working on a laboratory project when he realizes that to move further in his investigations he will need to know more mathematics. The laboratory will help to make it clear that mathematics is relevant, a handle on the real world; a laboratory central to the course enables us to teach mathematics in a real-world context. Further,

---

Education Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts.

the laboratory will demand of the student that he formulate and test models of physical, chemical or biological systems and thereby develop a facility in constructing such models. Finally, because laboratory experiments will be available in a variety of fields, the general usefulness of mathematics will be apparent, and this will mitigate the all too prevalent belief that mathematics is a set of tricks, each invented to solve a specific problem and useful nowhere else.

The inclusion of computers in the course is also motivated by several considerations. Computers are here to stay; they will increasingly affect the formulation and analysis of intellectual problems in every field. Computers, more than any other single thing, impose upon students (and no less upon trained professionals) the need for logical thinking and for explicitness and precision. But most important, computers take the drudgery out of mathematics and thereby make it possible to attack real problems rather than the uninteresting ones forced upon us when we are compelled, for the sake of calculational simplicity, to restrict ourselves to linear, quadratic, and (occasionally) cubic laws. I think all of us have many times had the experience of sitting down to write an examination or a problem set with two noble criteria in mind: (1) the problems should be realistic, and (2) they should be interesting. In a matter of minutes we toss out both criteria in favor of two others far less noble: (1) the problems must take a finite amount of time, and (2) they must take a finite amount of paper. The computer makes it possible to re-establish and fulfill the criteria of reality and interest.

Finally, the course is one based on calculus. This choice reflects the fact that the purpose of calculus is to analyze *change* and change is the concept that must be grasped and worked with by anyone who wishes to understand and make use of natural phenomena. This is why calculus is fascinating and why we feel it must be presented to the student without making him first pass through the often dull prerequisites that obliterate his natural enthusiasm and can deter him from further study rather than prepare him for it.

With these general comments in mind, I would like to describe some of the laboratory and the computer aspects of this course. The design and development of most of the laboratory is the work of my colleague William Walton. We have purposely kept the laboratory equipment as simple and as inexpensive as possible in order that the course be feasible in other schools. The piece of laboratory equipment that we think gives the most for our money is the accelerometer shown in Figure 1(a). It is merely a piece of flexible, transparent tubing filled with colored alcohol. The alcohol has a small bubble in it which moves from its equilibrium position when the device is subjected to acceleration. Using the earth's gravitational field, the accelerometer is easily calibrated.

Last summer some students took this instrument for a subway ride,

(a)

(b)

FIGURE 1 (a) Accelerometer made from plastic tubing filled with colored alcohol, (b) data collection.

measuring the acceleration as a function of time between Kendall Square and Harvard Square [Figure 1(b)]. A graph of their data is shown in Figure 2. By means of numerical integration they obtained the velocity of the train as a function of time as shown in Figure 3 from which we see that the maximum speed attained is about 45 mph. By a second numerical integration the students determined position as a function of time as shown in Figure 4. In the lower right-hand corner of this figure are the results of their calculations. As we see, their predictions are quite accurate.

To anyone who regards this as nothing more than an entertaining game I would point out a few things. First of all, the student has learned something about integration—what an integral is and how it may be evaluated, even

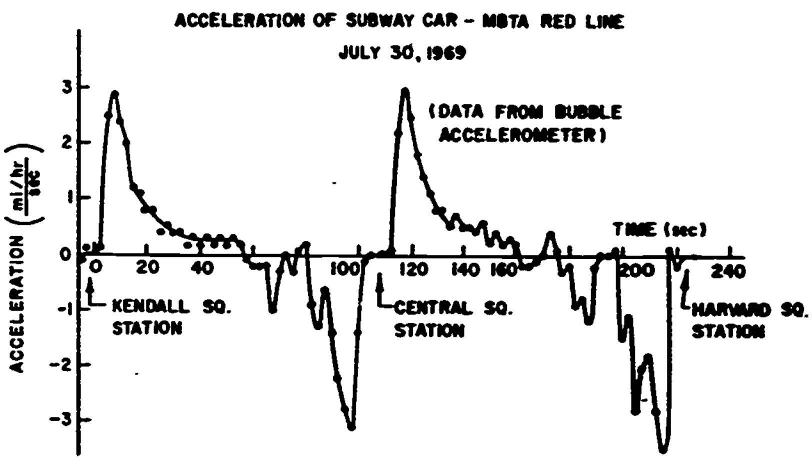


FIGURE 2 Acceleration vs. time. Data taken by students on subway ride from Kendall Square to Harvard Square, Cambridge, Massachusetts.

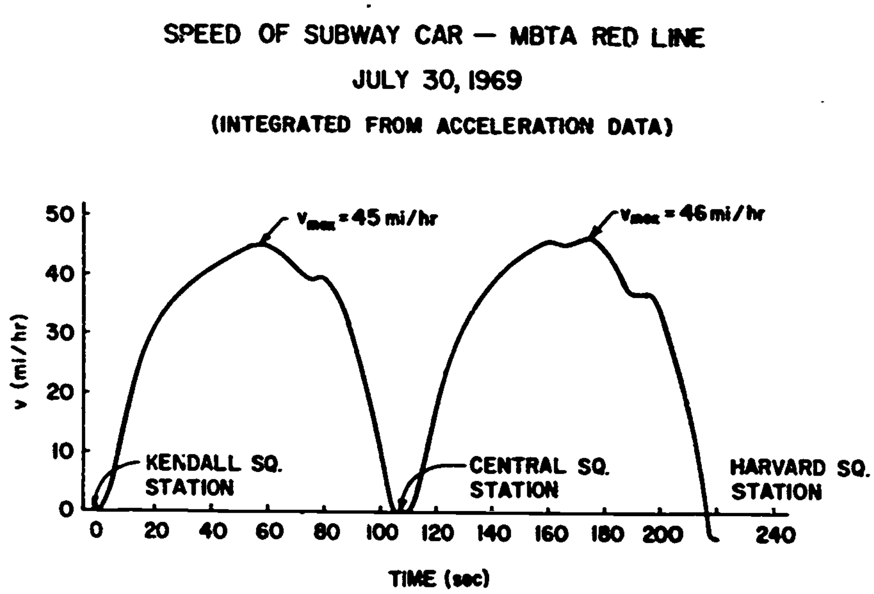


FIGURE 3 Velocity vs. time found by numerical integration of data in Figure 2.

though one has no convenient formula for the integrand. Secondly, the relations among position, velocity and acceleration become very clear from this experiment, and these three quantities are paradigms of function, first derivative and second derivative. Finally, when you ask how a submarine is

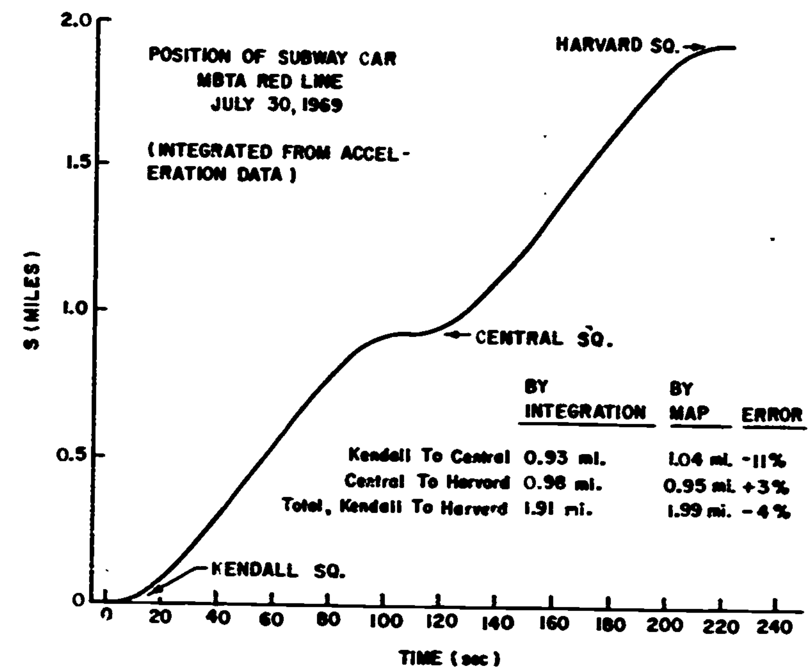


FIGURE 4 Distance vs. time found by numerical integration of data in Figure 3.

navigated under the polar ice cap or how a rocket finds its way to the moon, you soon realize that you have been dealing in this innocuous experiment with a crude but effective inertial guidance system. This is all the more impressive when you recognize that the accelerometer can be built for about thirty cents, so that the major expenditure involved in this experiment is for subway fares.

With this illustration of one of our laboratory projects, I would like to turn to the other main feature of the course we are developing, the use of computers. We are by no means the only ones to recognize the potentialities of the computer in teaching calculus. Nonetheless, I would like to present some examples of computer problems that we have incorporated in our work, many of which have already been used by students. Before doing this, however, let me express a bit of my philosophy in regard to the use of computers in general and in teaching in particular.

To begin with, it is my firm belief that for most teaching purposes the computer should be used in the time-sharing mode. There are two reasons for this. The first is the greater accessibility of computer facilities in the time-sharing mode. The user does not have to wait upon the whim of an operator; does not, usually, need to punch, correct and retain huge stacks of cards; and does not run the risk of having his program pushed further and further down the list as programs with higher "priority" are brought to the computer room.

The second reason for preferring the time-sharing mode is that each student has what is, effectively, his own computer. Errors can be corrected on the spot with practically zero turn-around time, and the student's communication with the computer is, conveniently, through a typewriter keyboard with the possibility of storing or taping programs for later use.

Another of my convictions is that the computer language with which the user "speaks" to the machine should be as simple as the genius of man can make it. To me this means BASIC. This is the language I prefer above all others, probably because it is the only one I have ever succeeded in learning. This is not a plea for a simple language but a cheer for one that has already been written, BASIC, and a recommendation that it or one of comparable simplicity be used whenever the computer is utilized in a pedagogical role. It may be true that BASIC is not so elegant as some languages, nor so flexible, but for teaching calculus elegance and flexibility are superfluous. The student should be able to learn all he needs to know about computing as quickly and easily as possible, for his real learning task is elsewhere.

Finally, students should write computer programs themselves, with pre-written programs avoided as much as possible. I do not mean by this to condemn student interactive programs in which the student and the machine engage in a dialogue. These, of course, are pre-written and play a useful role. But in the case of problems to be solved by the student using a computer, I advocate giving as little advice and help as possible as far as writing the program is concerned. The student stands to gain a great deal in thinking through the logic for himself, and for stimulating logical thinking, I believe this experience is second to none that can be provided in a classroom-laboratory context.

I turn now to some specific computer problems in calculus which we have used or intend to use in our course. Although I cannot recall having lifted any of these examples from other sources, I shall not guarantee that none will look familiar. Nor are they necessarily "clever" problems; throughout we are guided only by the wish to aid the student in his learning process. Also, like the course itself, the topics covered in these problems range from elementary to advanced; and I shall discuss these examples in order of increasing sophistication.

The first few examples illustrate the idea of limit. This concept appears to trouble many students who go through calculus with no clear understanding of what a limit is, though they grow very proficient in the purely mechanical processes of differentiation and integration. An excellent way to overcome this problem is to have the student use the computer to calculate a limit and thereby face up to the fact that it is a *number* which can be arrived at by a series of *arithmetic* operations. Figure 5 shows such a program and its results. The student is given an expression for the area of a 2^{m+1}-sided polygon circumscribed about a circle of unit radius in terms of one of 2^m sides. It is best if the student can derive this expression or at least see it derived, but it is

$$A_{m+1} = 2A_m \,/\, \left[1 + \sqrt{1 + (A_m/2^m)^2}\right]$$

```
10 LET Q=4
20 LET A=4
30 PRINT A
40 LET R=1+(A*A)/(Q*Q)
50 LET A=2*A/(1+R↑.5)
60 PRINT A
70 LET Q=2*Q
80 GOTO 40
90 END
```

```
4
3.31371
3.1826
3.15172
3.14412
3.14222
3.14175
3.14163
3.1416
3.14159
3.14159
3.14159
3.14159
3.14159
```

FIGURE 5 Calculation of the ares of a 2^m-sided polygon circumscribed about a unit circle, showing the area approaching π as m increases.

not essential. He can in any case be asked to compute the limit of this area as m approaches infinity. In practical terms, of course, this means he finds the areas of polygons with 4, 8, 16, 32, etc. sides. The resulting limit is a number, and one he is sure to recognize. But the best feature of this is its immediate significance geometrically; it is intuitively clear that as the number of sides increases, the series of polygons thereby generated tend to "look" more and more like the unit circle. It is quite natural then that their areas should "look" more and more like that of the unit circle, namely π. This exercise may be most effective if the student is urged to do a little drawing before he does the calculation to see if he can *predict* the limit. Such a procedure does two things: it helps to develop a strong intuitive feeling for the concept of limit and at the same time brings home the point that it is a well-defined concept leading to a *number*.

Another example of limit, a somewhat more sophisticated one having an interesting consequence, is represented in Figure 6. The function $e(x) = (1 + x)^{1/x}$ is evaluated for smaller and smaller values of x. The program that does this calculation, as well as the printout are shown in the Figure. The succession of x's are 1, 1/2, 1/4, 1/8, etc. We recognize that this limit is Euler's number e, and about halfway through the list of numbers we see the familiar numerical value correct to four figures, about one part in 4000. But then what happens? After obtaining this accurate value four times in succession we find that the calculation begins to deviate away from it, ending

$$e(x) = (1 + x)^{1/x}$$

```
10 LET X=1
20 LET E=(1+X)↑(1/X)
30 PRINT E
40 LET X=X/2
50 GOTO 20
60 END
```

```
2
2.25
2.44141
2.56578
2.63793
2.67699
2.69734
2.70774
2.71299
2.71564
2.71695
2.7176
2.7176
2.7176
2.7176
2.70119
2.69388
2.70851
2.65047
2.59368
2.37841
2
2
4
.0625
3.90625E-03
```

FIGURE 6 Numerical evaluation of the limit, as x approaches zero, of $e(x) = (1 + x)^{1/x}$.

finally in gibberish. Far from being a disaster, this behavior presents a fine opportunity to discuss limits in some depth.

The function $e(x)$, at first glance, might conceivably approach either 1 or infinity as x tends to zero. It settles down on something between 1 and infinity because of the delicate balance between the rate at which $(1 + x)$ approaches one and the rate at which $1/x$ approaches infinity. As x gets smaller and smaller, the computer, which represents a number by a finite number of digits, begins rounding off pieces of the x when it forms the combination $(1 + x)$. It may also have difficulties in exponentiating when $1/x$ is very large. Thus, for very small values of x, the delicate balance is upset and the limit lost. The student can investigate this matter for himself by calculat-

```
  sin x
  -----
    x

10 LET X=1
20 LET R=(SIN(X))/X
30 PRINT R
40 LET X=X/2
50 GOTO 20
60 END

 .841471
 .958851
 .989616
 .997398
 .999349
 .999837
 .999959
 .99999
 .999998
 .999999
 1
 1
 1.
```

```
 1 - cos x
 ---------
     x

10 LET X=1
20 LET R=(1-COS(X))/X
30 PRINT R
40 LET X=X/2
50 GOTO 20
60 END

.459698
.244835
.124351
.062418
3.12386E-02
1.56174E-02
7.79724E-03
3.89099E-03
1.95312E-03
9.15527E-04
3.66211E-04
2.44141E-04
0
0
0
-7.81250E-03
0
-.03125
-.0625
0
-.25
0
0
0
-4
-8
-16
-32
-64
-128
-256
```

FIGURE 7 Numerical evaluation of the limits, as x approaches zero, of $(\sin x)/x$ and $(1-\cos x)/x$.

ing the limit as x tends to zero of $(1+x^2)^{1/x}$ and $(1+x)^{1/x^2}$ in which the balance has been upset in advance, as it were. This careful examination of the anatomy of a limit is really quite sophisticated; nonetheless, through the use of computers it is accessible even to the beginner.

The next pair of examples are again calculations of limits but from a somewhat different point of view, for both are examples of indeterminate forms. The importance of such forms is usually overlooked, I think. They are taken up and discussed in conjunction with l'Hôpital's rule or Taylor series at a fairly late stage in the first calculus course. But the existence of indeterminate forms and the need to deal with them is one of the reasons for inventing limits—and therefore calculus itself. Thus an examination of indeterminate forms at an earlier stage seems to me to be at least advisable.

The two indeterminate forms shown in Figure 7 are important in and of themselves for both are involved in showing that the cosine is the derivative of the sine. The first is the limit as x approaches zero of $(\sin x)/x$. Ordinarily this limit is handled with a curious little geometric argument which often leaves students unsatisfied and unconvinced. The computer gives the result with no difficulty. I would not stop at this point, for it is vital that the student know not only the value of the limit, but why it is this value. A few pictures comparing arc lengths and chords can make it reasonable that $\sin x$ and x are nearly equal if x is small enough. In a sense, this explains the result the computer has found and points the way to a study of Taylor series.

The other of the two indeterminate forms shown in Figure 7 is the limit as x approaches zero of $(1-\cos x)/x$. Note that the correct answer (zero) is printed out several times and then lost because of the truncation and round-off error which arise in taking the difference $1-\cos x$ when x is small. Often, when obviously wrong answers arise this way, the student is prompted to ask how one can ever hope to make accurate calculations in the face of the damage round-off error can do, and once more we have a natural lead-in to Taylor series.

I turn next to a description of some computer problems in integral calculus. Here I will put aside the obvious possibilities of evaluating integrals as Riemann sums with many terms, except to mention that the computer provides a means of testing various numerical integration methods (trapezoidal rule, Simpson's rule, etc.). Perhaps more important, it can help dispel the idea that a definite integral exists and has a value only if you can find the antiderivative of the integrand and apply the fundamental theorem of calculus. Among the antidotes for this poison are a calculation of a short table of the error function, or a subway ride with an accelerometer.

In dealing with a computer in teaching integral calculus, one rapidly becomes aware of its value as a theorem verifier. There are certain theorems in integral calculus which are usually not proved in elementary courses; they are simply taken on faith, and verification in a variety of cases can be extremely useful. Indeed, the evaluation of Riemann sums for many terms verifies the generally unproved assertion that the limit in terms of which a definite integral is defined actually exists.

The first example of computational methods in integral calculus verifies another assertion. We consider the standard definition of the definite integral:

$$\int_a^b f(x)\,dx = \lim_{\substack{\max \Delta x_k \to 0 \\ n\to\infty}} \sum_{k=0}^{N-1} f(x'_k)\,\Delta x_k$$

The assertion is that x'_k may be *any* value of x in the interval Δx_k, the limit being independent of this choice. To verify this we use the simple integral

$$1/3 = \int_0^1 x^2\,dx \cong (1/N) \sum_{k=0}^{N-1} (k+t)^2/N^2$$

The integral can be approximated by the sum as indicated, and when N, the number of subintervals, is large this sum should be approximately 1/3 the value of the integral. The parameter t which appears here varies from 0 to 1, and as it does so, x_k' varies from x_k on the left end of the interval to x_{k+1} on the right. Figure 8 shows the results for five different values of t: 0, .25, .50, .75 and 1.00. There is little question that the theorem is valid in this case—it really does not matter where we put x_k'.

```
10  DIM X[300]
20  DIM Y[300]
30  DIM Z[300]
40  INPUT X[2],X[3],X[4],X[5]
50  LET X[1]=0
60  LET X[6]=1
70  LET Z=5
80  LET M=1
90  FØR P=1 TØ 6.
100  NEXT P
110  FØR K=1 TØ Z
120  LET Y[K]=.5*(X[K]+X[K+1])
130  NEXT K
140  LET S=0
150  FØR L=1 TØ Z
160  LET S=S+(Y[L]↑2)*(X[L+1]-X[L])
170  NEXT L
180  PRINT S
190  IF M=5 THEN 320
200  FØR J=1 TØ Z+1
210  LET Z[2*J-1]=X[J]
220  NEXT J
230  FØR J=1 TØ Z
240  LET X[2*J]=Y[J]
250  NEXT J
260  LET M=M+1
270  LET Z=2*Z
280  FØR J=1 TØ Z+1 STEP 2
290  LET X[J]=Z[J]
300  NEXT J
310  GØTØ 110
320  END
```

```
?.2,.4,.6,.8
 .33
 .3325
 .333125
 .333281
 .33332

?.13,.37,.61,.88
 .329062
 .332266
 .333066
 .333267
 .333317

?.23,.35,.59,.93
 .32772
 .33193
 .332983
 .333246
 .333311

?.68,.73,.9,.98
 .306668
 .326667
 .331667
 .332917
 .333229
```

FIGURE 8 Numerical evaluation of an integral verifying that the value is independent of where in each subinterval the integrand is evaluated.

Another theorem pertaining to definite integrals asserts that the interval of integration may be subdivided arbitrarily, the limit of the sum being independent of this choice of subdivision. I have been able to think of three different ways to illustrate this theorem numerically, and I have selected for purposes of illustration one in which the interval is divided into five parts. The end points of the interval are fixed, of course, but the other four values of x_k can be selected at random and entered from the keyboard as input. The machine then evaluates the sum; subsequently it will halve each interval and re-evaluate. It does this a total of four times. The program is shown on Figure 9 where the example used is $\int_0^1 x^2\,dx$. Some results are also shown here. In the first my choices for the x_k are a sane and civilized .2, .4, .6 and .8. The sum clearly converges to the value 1/3. In the next two cases vary the input somewhat, though not dramatically, and the sum again converges to 1/3. In

```
10  INPUT N,T
20  PRINT "T="T
30  PRINT "N","INTEGRAL"
40  LET S=0
50  FOR K=0 TO N-1
60  LET S=S+((K+T)*(K+T))/(N*N)
70  NEXT K
80  LET S=S/N
90  PRINT N,S
100  LET N=2*N
110  GOTO 40
120  END
```

T= .5

N	INTEGRAL
10	.3325
20	.333125
40	.333281
80	.33332
160	.33333
320	.333333
640	.333334

T= 0

N	INTEGRAL
10	.285
20	.30875
40	.320938
80	.327109
160	.330215
320	.331773
640	.332552
1280	.332943

T= .75

N	INTEGRAL
10	.358125
20	.345781
40	.33957
80	.336455
160	.334895
320	.334114
640	.333724
1280	.333529

T= .25

N	INTEGRAL
10	.308125
20	.320781
40	.32707
80	.330205
160	.33177
320	.332552
640	.332942
1280	.333138

T= 1

N	INTEGRAL
10	.385
20	.35875
40	.345938
80	.339609
160	.336465
320	.334898
640	.334115
1280	.333724

FIGURE 9 Numerical evaluation of an integral verifying that the value is independent of the partition of the interval of integration.

the last example I take the bull by the horns and enter some thoroughly outrageous values. The numbers shown, .68, .73, .90 and .98 are, to within a factor of 100, the opus numbers of the four symphonies of Brahms. The sum converges to 1/3.

I move on now to some topics that are more advanced or, in any case, have long been regarded as advanced. Here we have examples that illustrate one of the most important advantages of using computers in teaching calculus, namely that it enables us to discuss certain things at a much earlier stage than would otherwise be possible. In my judgment, the most interesting and useful part of calculus is the field of differential equations. It is sad but true, however, that it is often barely alluded to in a first course in calculus and students generally do not come to grips with the concept of differential equations, let alone methods of solving them, until their second year at the earliest.

An especially fine way to introduce differential equations to students early is via numerical considerations. The benefits to be reaped through this are great. First, it puts into the student's hands what may be the single most useful mathematical concept he will ever have at his disposal, and this can make a big difference to him in the breadth and depth of understanding open to him in physics, biology, psychology, ecology, electrical engineering, or whatever may be his field of interest; and it does this early in his education so that less postponing and marking of time will be necessary in other parts of his training while waiting for him to gain more mathematical expertise. Second, by starting young, he grows up with differential equations, is comfortable with them sooner, and grows proficient sooner and more easily in recognizing those problems that require differential equations for their solution. Third, numerical procedures provide him with a means to solve differential equations other than the endless list of methods scrupulously given in every text and guaranteed to work only on the problems at the end of the chapter. This means he can spend more of his time where he should: in learning how to set up differential equations wherever they are called for in his work. Finally, and perhaps most important, the subject of differential equations is the meeting place for all of calculus, differential and integral, and is an ideal place to consolidate one's knowledge of the subject.

To illustrate what can be done with differential equations and computers, we consider an eigenvalue problem. A string of variable density and unit length, fixed at both ends is described by the equation

$$d^2u/dx^2 = -k^2[1 + cx(1-x)]u(x)$$

with $u(0) = u(1) = 0$. The constant c is a given parameter. When its value is

```
10  INPUT C,N,K
20  LET U0=0
30  LET U1=1/N
40  FOR J=2 TO N
50  LET A=1+(C*J*(1-J/N))/N
60  LET U2=(2-((K*K)/(N*N))*A)*U1-U0
70  LET U0=U1
80  LET U1=U2
90  NEXT J
100 PRINT U2
110 END

?0,50,2           ?0,50,3.15         ?1,50,2.8
.454703           -2.83282E-03       1.73062E-02

?0,50,3           ?0,50,3.14         ?1,50,2.9
4.69301E-02       3.30716E-04        1.87351E-02

?0,50,4           ?0,50,3.143        ?1,50,2.85
-.189531          -6.02365E-04       -1.02086E-03

?0,50,3.5         ?0,50,3.141        ?1,50,2.83
-.100475          3.24845E-05        6.24170E-03

?0,50,3.1         ?1,50,3            ?1,50,2.84
1.32547E-02       -5.23091E-02       2.60010E-03

?0,50,3.2         ?1,50,2            ?1,50,2.845
-1.84247E-02      .36927             7.95290E-04
```

FIGURE 10 Numerical solution of an eigenvalue problem.

zero the equation then represents the more familiar case of a string of uniform density. The constant k, of course, is the eigenvalue.

The program for integrating this equation is shown in Figure 10 along with some results. The three numbers following the question mark are c, N (the number of mesh points) and the guessed value of k. Directly below each set of these three numbers is the corresponding value of the solution at the end poin 1. When k is correctly guessed this number will be zero. The game, therei. , is one of iterating to the value of k which will make that number as small as possible. The first set of runs is for the case $c = 0$, i.e., a string of uniform density, for which the lowest eigenvalue is known to be π. As you can see the best value obtained here with 50 mesh points is 3.141. There is also a run shown here for $c = 1$ (for which the best eigenvalue is about 2.845). A similar program was written and used by a student who, prior to doing this work, had never heard of eigenvalue problems and was certainly in no position to handle this differential equation, or most others, by standard methods.

It is in the context of numerical solutions of differential equations that one can best answer a frequently heard criticism of a computer approach to calculus. This criticism expresses the fear that in dealing with calculus numerically the student will not be exposed to, and will therefore not learn, analytic techniques. For despite the pedagogical advantages of computers, and despite their growing utilization in all scientific and technological fields, analytic techniques will continue to play a central role. The answer to this criticism is that a student using a computer will eventually learn just as much, if not more, of analytic techniques as the student whose training has been along traditional lines.

The student who calculates an integral numerically wants a feeling of confidence in his results and may wish to compare several schemes for numerical integration. The obvious procedure is to try these schemes on an integral whose value can be found analytically, and this means learning some of the techniques of integration. (Unlike his teacher, incidentally, the student is not content to deal with $\int_0^1 x^2\,dx$–the messier the integral, the more fun it is, especially if he gets the right answer). Another example is that of the student who wants to know how the computer is smart enough to find numerical values of sines, cosines, exponentials, logarithms, etc. He is ripe to learn about Taylor series.

Nowhere better than in the case of differential equations do we see the student prompted to learn analytic methods as a result of his computer work. They learn about series solutions to start a numerical integration; and about asymptotic forms to fit boundary conditions at infinity; after several days of struggling to solve a second-order equation, the student looks through the book and discovers the transformation that eliminates the troublesome term in the first derivative; and so on. The student who first worked on the eigenvalue problem worried about the accuracy of his solutions. I pointed out that for small deviations from the uniform density case his solutions could be checked using perturbation methods. He learned a little about this, and I am sure he would have learned more but for the fact that our academic year suffered a sudden and early death resulting from problems that cannot be solved on a computer.

The usefulness of a computer is in direct proportion to the amount of thinking you do before you begin to program. I would like to predict that a generation trained in mathematics with the aid of computers will know more, not less, of analytic methods than those trained in the traditional way.

I hope that these comments give some idea of how we are using computers in teaching calculus. Among the advantages of this method are:

1. Computers require and therefore tend to stimulate logical, precise thinking.
2. Computers are highly useful in making concepts, such as that of the limit, firm; they can help to dispel the idea that such concepts are inherently ill-defined.

3. Computers enable us to provide realistic and interesting problems by taking the drudgery out of the work.
4. Computers provide a means of verifying theorems which are not ordinarily proved in beginning courses.
5. Computers enable us to arrive at certain concepts at an earlier stage in the student's work and provide him with valuable mathematical tools sooner than he might otherwise have them.
6. Computers at one and the same time provide both freedom from analytic methods and a climate in which the student is prompted to learn those methods.

No doubt there are many other advantages as well, which we will see as we continue our work. Admittedly there are disadvantages too. Perhaps the most serious is the tendency of some students all of the time and all students some of the time to hide behind the computer—it's very easy to sit at a terminal typing away and give yourself (and a gullible world) the impression that you are busy accomplishing much that is worthwhile, when all you are really doing is playing games. Less serious, but more frustrating, is an almost universal desire on the part of students (or anyone using a computer, for that matter) to write and rewrite programs making them more and more sophisticated and more and more clever. Very few students with whom we have worked, for example, are content until they have written themselves out of the program, leaving it completely automatic. While this is desirable in a long program that will be run on a production basis for the next five years, it is scarcely necessary for a nine-line code which evaluates $\int_0^1 x^3\,dx$ with various choices of subintervals to be polished to a lapidary finish. The student will have wasted several days of his time (to say nothing of several hours—expensive hours—of computer time) in bringing this finger exercise to a pitch of perfection it can hardly warrant.

Nonetheless, in our judgment the advantages of the computer in teaching calculus far outweigh the disadvantages. We believe that approaches such as the one we are developing, involving a laboratory and extensive use of computers, will bring mathematics to many who need it but have never before had the opportunity to learn it. Much remains to be done; we have only made a beginning. But we believe the impact of these methods on teaching mathematics will be enormous.

The Computer in the Calculus Course

WARREN STENBERG

Computer-oriented calculus has gained many adherents over the last few years. There are at least five computer-flavored calculus texts on the market, and numerous colleges have prepared computing materials to supplement conventional calculus texts. In some computer-related approaches the calculus is merely presented with a strong dose of numerical analysis so as to be "cognizant of the existence of the computer." Other approaches involve the traditional approach to the presentation of theory but include computer-based illustrations and programming assignments in supplementary materials.

THE COURSE

The CRICISAM calculus text, which I shall discuss, makes a much more radical departure; the very concepts of calculus are given an algorithmic treatment. This entails a change in the order of topics, in the formulation of axioms and definitions, in the proofs of theorems and in the emphasis given various topics. The student entering this one-year course in functions of a single variable emerges having covered all the theory and most of the applications of the traditional treatments, but the goal is reached by a very different route. Some of these differences are enumerated below:

I. Order of Topics
   - A. Sequences and definite integrals (first two chapters)
   - B. Uniform continuity (introduced prior to pointwise continuity)
   - C. Numerical solution of differential equations, with error bounds

II. New or More Heavily Emphasized Topics
   - A. Numerical integration (mid-point, trapezoid, Simpson's)
   - B. Newton's method
   - C. Convexity
   - D. Differential equations

Department of Mathematics, University of Minnesota, Minneapolis, Minnesota.

III. New Approaches to Theory
   A. Completeness axiom
   B. Definition of definite integrals
   C. Existence theorems proved algorithmically

We have avoided trying to do everything by algorithmic means. Thus, algorithms are almost completely absent from the treatment of differentiation. To bring them in would have been more obfuscatory than illuminating.

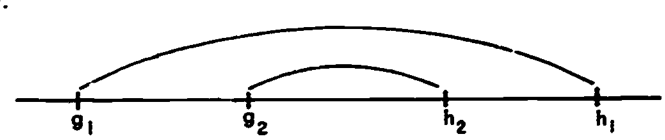
The reason we feel that this algorithmic treatment of calculus can be more effective than traditional treatments is that it provides the student with a dynamic or "hands-on" computational experience which makes the concepts of calculus come to life and have new significance. Since these volumes* are now available to the public, readers interested in a complete discussion of the course outlined above can find it by studying them. Therefore, I shall try here to present briefly the approach and thrust of the course by discussing some examples. Since communication with students is of primary concern, this material is presented essentially as it is put before the student. Although the first two examples are more properly algebraic rather than analytic, they form a useful introduction to concepts of recursive procedures and convergent sequences.

EXAMPLE 1. THE SQUARE ROOT ALGORITHM

If two pairs of positive numbers (g_1,h_1) and (g_2,h_2) have the same product

$$g_1 \times h_1 = g_2 \times h_2$$

then the pair containing the smallest of the four numbers must also contain the largest. That is, if the numbers are represented as points on the line, the segment joining one pair contains the segment joining the other, as shown below.



In particular, if

$$g_1 \times h_1 = \sqrt{a} \times \sqrt{a}$$

Calculus, A Computer–Oriented Presentation, Center for Research in College Instruction of Science and Mathematics, Florida State University, Tallahassee, 1969.

we see that since the two factors on the right coincide, $\sqrt{a}$ must lie between g_1 and h_1.

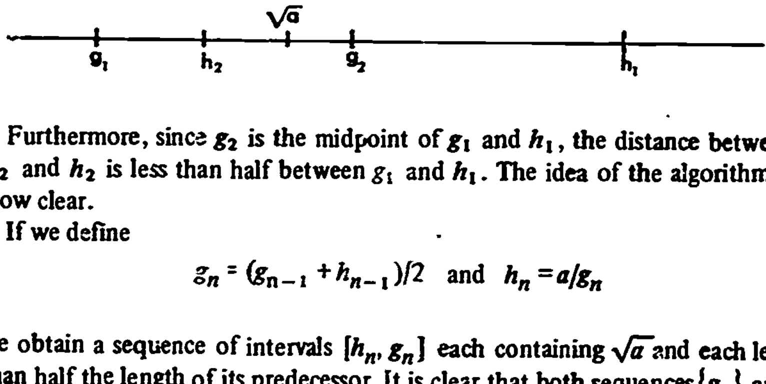
This idea constitutes the basis of an algorithm for approximating square roots to any desired degree of accuracy. Suppose that g_1 is chosen arbitrarily and that $h_1 = a/g_1$; then $g_1 \times h_1 = a$ and $\sqrt{a}$ lies between g_1 and h_1. Now let

$$g_2 = (g_1 + h_1)/2 \quad \text{and} \quad h_2 = a/g_2$$

Since g_2 is the midpoint of g_1 and h_1, it lies between g_1 and h_1. Moreover, since

$$g_2 \times h_2 = a = g_1 \times h_1$$

we see that h_2 also lies between g_1 and h_1 and moreover $\sqrt{a}$ lies between g_2 and h_2 as shown below.



Furthermore, since g_2 is the midpoint of g_1 and h_1, the distance between g_2 and h_2 is less than half between g_1 and h_1. The idea of the algorithm is now clear.

If we define

$$g_n = (g_{n-1} + h_{n-1})/2 \quad \text{and} \quad h_n = a/g_n$$

we obtain a sequence of intervals $[h_n, g_n]$ each containing $\sqrt{a}$ and each less than half the length of its predecessor. It is clear that both sequences $\{g_n\}$ and $\{h_n\}$ converge to $\sqrt{a}$ ("convergence," though not yet defined, is used here intuitively). Note that in Figure 1 we do not use subscripted variables, however, the values of g and h on the nth line of output are g_n and h_n. The program begins with initialization of the variables with g arbitrarily chosen equal to unity and includes a test which terminates the program when $|g_n - h_n| < \epsilon$, since we know that $\sqrt{a}$ must lie between them.

We have insidiously introduced the notion of ϵ as an error control without terrifying the student. After an additional example and some discussion we find that this notion leads to the following definition of convergence:

> If, for every positive number ϵ, there is an integer N sufficiently large such that a_n differs from L by less than ϵ whenever $n \geqslant N$, then we say that the sequence $a_0, a_1, a_2 \ldots$, converges to L.

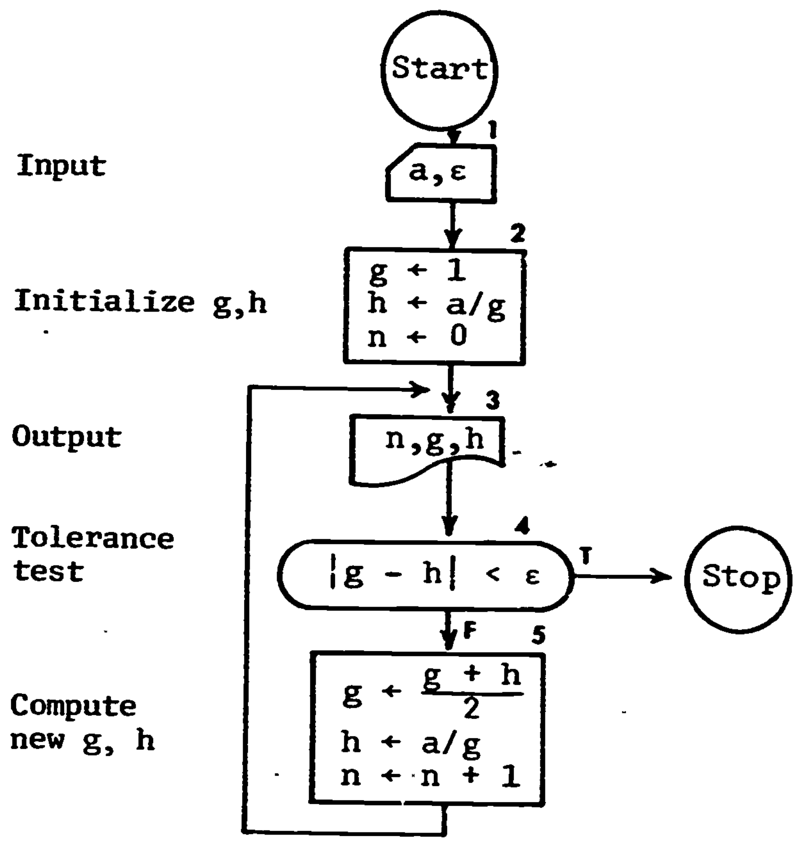


FIGURE 1 Flow chart for square root algorithm.

An interesting footnote to this example is that the student finds that convergence is strikingly faster than predicted, a fact which serves to further motivate his interest in numerical analysis.

We then proceed to a discussion of error, noting that the error at the $(n+1)$st stage, $E_{n+1} = g_{n+1} - \sqrt{a}$, is

$$
\begin{aligned}
E_{n+1} = g_{n+1} - \sqrt{a} &= (g_n + h_n)/2 - \sqrt{a} \\
&= (g_n + a/g_n)/2 - \sqrt{a} \\
&= (g_n^2 - 2\sqrt{a}\,g_n + a)/2g_n \\
&= (g_n - \sqrt{a}\,)^2/2g_n
\end{aligned}
$$

Thus the error at the $(n+1)$st stage is on the order of the square of the error at the nth stage.

Confining ourselves to the case $a > 1$ and $g_0 = 1$ so that all future $g_n > 1$, we see that

$$E_{n+1} \leq E_n^2/2$$

Hence, if we reach the stage where $E_n < 1/10$, at the next stage the error will be less than 1/200. We list a few successive error bounds starting with 1/10:

1/10
1/200
1/8000
1/6400000000
1/81920000000000000000

This last value is very small indeed. We see that instead of merely halving the error at each stage, the order of magnitude of the error is halved each time through the loop.

EXAMPLE 2. FIBONACCI RATIOS AND THE COMPLETENESS AXIOM

The well-known Fibonacci sequence is defined by the recursion relation

$$a_{n+1} = a_n + a_{n-1}$$

and the initial conditions

$$a_1 = a_0 = 1$$

The first few terms of this sequence are 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 87, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, ... Clearly, this sequence does not converge, but if we consider the sequence of ratios

$$r_n = a_{n+1}/a_n$$

it is not immediately clear whether this sequence $r_1, r_2 \ldots$ converges or not. However, we can quickly calculate the value to which this sequence converges, if it does indeed converge, as follows:

$$\begin{aligned} r_{n+1} = a_{n+2}/a_{n+1} &= (a_{n+1} + a_n)/a_{n+1} \\ &= 1 + (a_n/a_{n+1}) \\ &= 1 + (1/r_n) \end{aligned}$$

Hence, if r_n converges to L, by our sum and reciprocal theorems for limits (by now proven) we have

$$L = 1 + (1/L)$$

so that

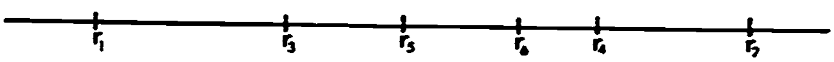
$$L^2 - L - 1 = 0; \quad L = (1 \pm \sqrt{5})/2$$

Since a sequence of positive forms cannot have a negative limit the only possible value for L is $(1 + \sqrt{5})/2$. If we program this algorithm for generation of the sequence $\{r_n\}$ on a computer, it will be found that the sequence apparantly coverges quite rapidly to $(1 + \sqrt{5})/2$.

Let us try to prove that the sequence $r_0, r_1, r_2 \ldots$ converges: it is easily seen using the Fibonacci recursion relation that

$$r_{n+1} - r_n = (-1)^n/(a_{n+1} \cdot a_n)$$

These differences alternate in sign while their absolute values tend monotonically to zero. In other words, each term lies between its two immediate predecessors with differences between consecutive terms tending to zero as shown below.



Thus, the odd-numbered terms form an increasing sequence, the even-numbered terms, a decreasing sequence with the length of the segment joining r_n and r_{n+1} tending to zero. It seems quite evident that the interval $[r_n, r_{n+1}]$ shrinks to a point. But there is no way of showing that there is a number associated with that point. We require one more axiom, the *Completeness Axiom:*

> If $L_1, L_2, L_3 \ldots$ and $R_1, R_2, R_3, \ldots$ are reprectively increasing and decreasing sequences with the difference $R_n - L_n \to 0$ as $n \to \infty$, then both sequences converge and to the same limit.

This form of the completeness axiom is most unusual for calculus texts, but it arises quite naturally in the computer setting. Furthermore, it turns out to be ideally adapted for virtually all computer applications of convergence of sequences. For example, it aids in the location of zeros of functions by the bisection method and in estimating integrals by means of upper and lower sums.

EXAMPLE 3. THE DEFINITE INTEGRAL

Integrals are introduced by considering the problem of area under a monotonic curve. Based on the highly intuitive monotonicity and additivity properties of area and the formula for the area of a rectangle, we can delimit the area, A, between the lower and upper sums L and U as shown in Figure 2.

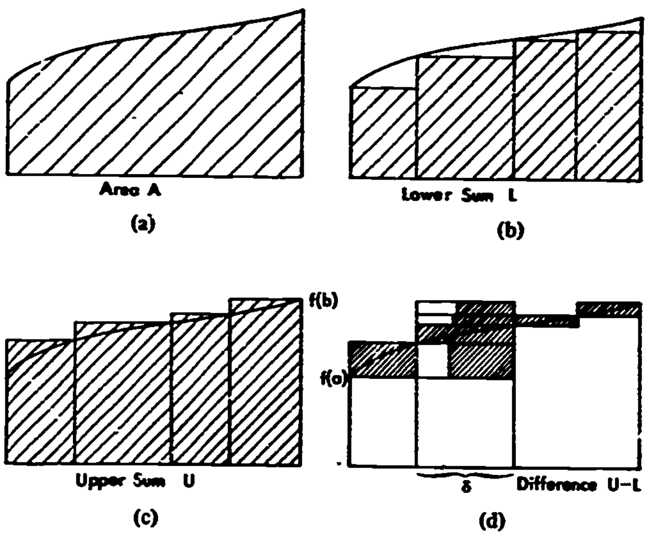


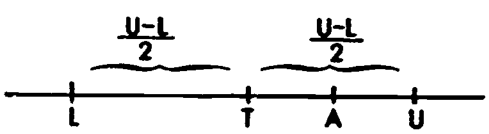
FIGURE 2 (a) Area (A) under the curve of a monotonic function $f(x)$ between the limits $x = a$ and $x = b$.
(b) A lower sum (L) approximation to area A.
(c) An upper sum (U) approximation to area A.
(d) The difference between U and L shown to be less than $\delta \cdot |f(b) - f(a)|$, where δ is the width of the widest subinterval.

So, letting

$$T = ½\,(L + U)$$

$$\therefore |T - A| < ½\,(U - L)$$

the numerical values being related as shown below for the area of Figure 2.



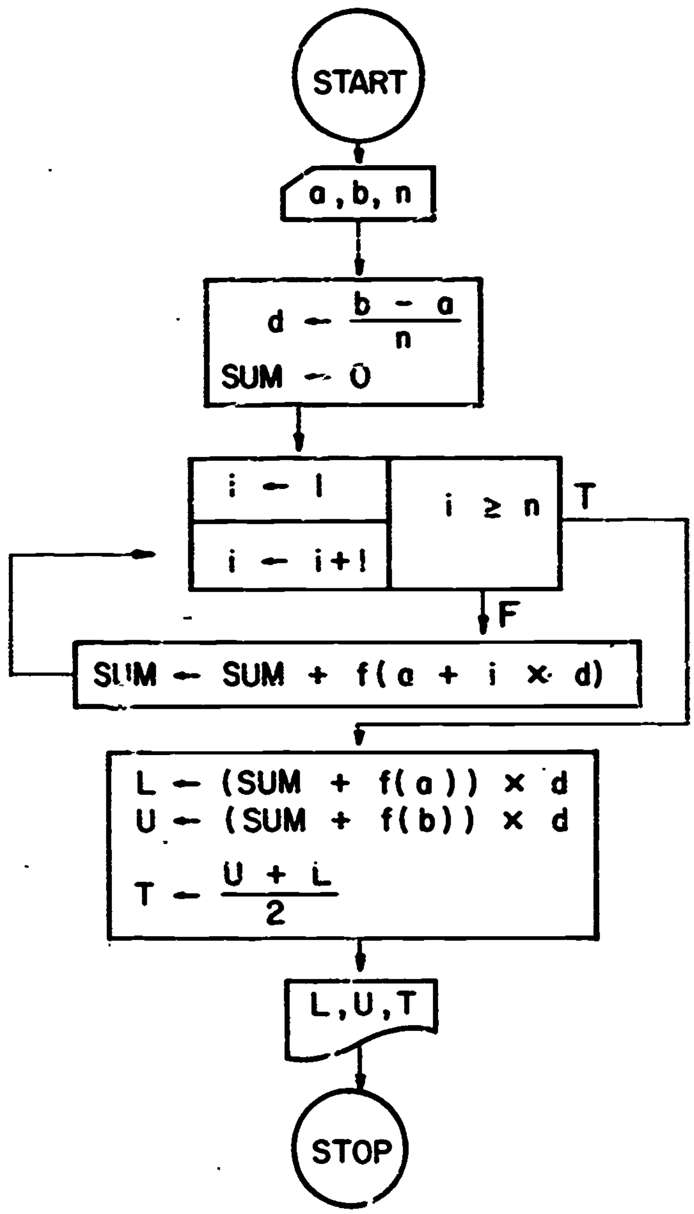


FIGURE 3 Flow chart for calculating the lower and upper sums for n subintervals in the interval $a < x < b$.

From Figure 2(d) we can see that the upper portion of the rectangle of width δ is larger than the sum of the (shaded) differences between U and L on each interval. Thus,

$$2|T-A| < U-L < \delta|f(b)-f(a)|$$

where δ is the width of the widest subinterval. In order to make the difference $|T-A|$ less than some ϵ, it suffices to choose

$$\delta < 2\cdot\epsilon/|f(b)-f(a)|$$

If the value of ϵ is small, the evaluation of T will involve much calculation but the computer algorithm is simple enough. We consider the case of equally spaced partition points and successively bisect the intervals to obtain sequences L_n and U_n of lower and upper sums converging to the desired area. Figure 3 shows the flowchart for this algorithm.

For non-monotonic functions we need a more general definition of upper and lower sums than is found in calculus texts; namely,

if $a = x_0 < x_1 < x_2 < \ldots < x_n = b$, and if

$m_i \leqslant f(x) \leqslant M_i$ for $x_{i-1} \leqslant x \leqslant x_i$ then

$$L = \sum_{i=1}^{n} m_i(x_i - x_{i-1}); \; U = \sum_{i=1}^{n} M_i(x_i - x_{i-1})$$

are defined to be the lower and upper sums of $f(x)$ over the interval $[a,b]$.

We do not mention the extrema of the function over the subintervals, since we do not have any prior knowledge of their existence, nor any way of finding them. In this treatment the function and the partition do not determine the upper and lower sums. But, surprisingly, this innovation simplifies the theory of integration, because the geometrical configurations representing upper and lower sums for a given partition also represent upper and lower sums for any refinement of the partition.

Thus, our resulting definition of the integral is:

> if there exist sequences of lower and upper sums $L_1, L_2, \ldots$ and $U_1, U_2, \ldots$ for the function $f(x)$ over the interval $[a, b]$ with $U_n - L_n \to 0$ as $n \to \infty$, then the common limit of these sequences is defined to be the area $A = \int_a^b f(x)dx$.

The student can see, from previous work, that sequences enjoying these properties must converge to a common limit and that any other pair of sequences satisfying the same properties converge to the same limit.

The above discussion gives at least a suggestion of how the theory of integration evolves from the algorithmic approach.

A One-Semester Introductory Physics Course

WILLIAM A. BOWERS

INTRODUCTION

Fitchburg State College is primarily a teacher training institution. In the spring of 1968, the physics department conducted a series of voluntary colloquies to determine the interest of high school students in T.V. taped presentations, air-track experiments, single-concept films and computers. The greatest interest was found to be in computers.

In the fall of 1968 funding for two small PDP 8/L computers was obtained by soliciting funds from local industries. The computers were obtained in January 1969, and approximately 100 high school teachers were given free instruction in programming. A computer-based physics course was conceived during this programming course, and a set of exercises was roughed out.

In the summer of 1969, twelve exercises were developed; (1) vectors, (2) speed, velocity, and acceleration, (3) projectile motion, (4) Newton's second law, (5) the spring, (6) the damped spring, (7) circular motion, (8) the inverse-square law, (9) the pendulum, (10) linear momentum (one dimension) (11) linear momentum (two dimensions) and (12) the rocket. The decision to develop a one-semester course from these experiments is due to the students' acceptance of the exercises and the comparative mathematical ease with which the material is developed. The need for such a course has long been evident at Fitchburg State College.

The one-semester physics course at Fitchburg State College is intended for non-science majors. The problem is to devise a means to show the students some of the ideas in physics and give them a chance to explore them in the laboratory. The computer is attractive, because it permits the student to investigate more parameters in the laboratory in a given time, it does routine arithmetic for him, it can produce graphs as solutions to problems and it can act as a simulator.

Last year, the twelve exercises were used as laboratory exercises for two-

---

Department of Physics, Fitchburg State College, Fitchburg, Massachusetts.

hour laboratories. The response of the students was encouraging, but in many cases more time was desired to allow them to investigate some of their own parameters. (Several times students came up with, "What would happen if . . .?"). One of the difficulties experienced was correlating the laboratory work with the class work, which was based on a standard text. This was tried in two instances by different instructors, and the conclusion of each was that the course would have been better if the text was used as a reference. A second difficulty arose in each case by not teaching programming and graphical analysis initially. An initial, complete treatment of programming and graphs would have been better than injecting bits of each as required. The students indicated this in their evaluation.

COURSE CONTENT

The course proper is preceded by three weeks of programming instruction in JOSS (FOCAL) and about one week of graphical methods. Since the computer has very limited core (about 750 available locations), economization of programs will be stressed. The programming instruction will be physics-oriented. This is followed by instruction on graphical methods, which will rely on the previous three weeks of computer programming. It will be used to investigate various types of functions such as the straight line, cosine, damped cosine, power curves, and logs.

The course proper begins in October with a development of finite-difference methods and their use as linear approximations. This is followed with the twelve exercises to be completed by Christmas, after which the students will complete and demonstrate projects begun after Thanksgiving.

The course, consisting of two, two-hour sessions per week, will be oriented toward laboratory investigations, with graphic solutions by the computer. The exercises will be presented at the rate of one per session. The laboratory investigations will be carried on by a group following the initial presentation. There will not be enough time available to complete the exercise in class, so each student will finish it as homework.

Finite-difference methods are developed through a definition of speed and acceleration using simple first-order approximations to position and velocity given by $s = s_0 + v_0(\Delta t)$, $v = v_0 + a(\Delta t)$, where ($_0$) refers to values at $t = t_0$, the beginning of the time interval. Initial speed is usually fixed at zero in the exercises. The desired time interval is determined, usually by measurement, and Δt is typically chosen to be 1/1000th of the total time. Position s_0 is measured in the laboratory, and the remaining quantity, acceleration, becomes the subject for investigation. The cases considered, for constant k, c, are acceleration, a, equal to: (1) k, (2) $-ks$, (3) $-kx-cv$, (4) $-k/x$, (5) $-k/x^2$, (6) $-k \sin x$.

The same general computer program is used in all examples. Graphical output is used in one-dimensional cases, and tabular output is used in two-dimensional cases. There is not enough core in the computer to store large numbers of variables, and, at present, our only graphing device is the teletype.

THE EXERCISES

A list of the exercises and a brief summary of each is given below:

1. Vectors. Compares experimental data from force table with graphical solutions and computed solutions.

2. Speed, velocity and acceleration. A ball is rolled across a rectangular table at constant speed in one direction and constant acceleration in the other. One-second markings are made in each case and projected to form a predicted plot for the combination. The table is then tilted and the ball given its original speed. (The exercise works pretty well if soft carbon paper is placed under the ball each time.)

3. Projectile motion. A spring gun is used. Three cases are investigated: (1) gun elevation and landing elevation are the same, (2) gun elevation higher than landing elevation, (3) gun elevation lower than landing elevation. Students compare maximum elevation and range with computed values. Computer output is graphic.

4. Newton's second law. Acceleration is determined for three cases of inclined planes: a ball on an aluminum track, a four-wheeled cart on a track and an air track. The time lapse in each case is about five seconds. The value for constant linear acceleration is determined, and experimental positions for each second are compared with computed positions. A discussion of the variations is used to introduce friction and rotational motion.

5. The spring. The spring constant and period for some mass are determined experimentally. These values are input to the computer which outputs a graphical solution for position, speed and acceleration. The speed curve is used to correct the period. The curves, with their computed maximum values, are used to obtain the relationships between position, speed and acceleration.

6. The damped spring. A dash pot, filled with water, is added to Number 5. Maximum excursions are determined experimentally in each direction. A constant c is adjusted until experimental and computed results agree. The damping constant is computed from experimental data and the decay envelope compared with previous data. The decay envelope is combined with the solution in Exercise 5 to obtain a solution to the problem.

7. Circular motion. The inverse-power law is developed from experimental date and graphs: log paper is introduced. Computer output is tabular. This exercise is under revision.

8. The inverse-square law. The procedure in Exercise 7 is reversed. If the change in speed varies inversely as the square of distance, what types of motion will result? A range of values is given for the initial investigation. Planetary motion is introduced. Computer output is tabular.

9. The pendulum. A simple pendulum is investigated. Students will write programs to determine the effect of increasing angle on period.

10. Linear momentum (one dimension). Momentum is introduced in one dimension (air track). Elastic and inelastic collisions are considered. The computer is used as a desk calculator to determine speeds from distance and time measurements.

11. Linear momentum (two dimensions). An air table and air pucks are used to study momentum in two dimensions. Students write their own programs for computations.

12. The rocket. Finite-difference methods are used to compute the trajectory of a rocket based on conservation of momentum with variable mass. A small rocket is fired and experimental data compared with computed data. The parameters are adjusted to fit experimental data and the rocket fired again. The process is repeated until the group can predict where the rocket will land.

Table Top Computers in Physics Teaching

JOHN M. GOODMAN

Computer usage can confer manifold benefits upon the educational process. Using a computer can help a teacher to better understand the material he is presenting. Students can also obtain greater insight into the subject matter of a course either by using a computer themselves or by examining the results of someone else's computer use. Numerical solutions of a physics problem whether performed by the student or merely presented to him with a discussion of the techniques used can materially lessen his anxiety over a new analytical technique for solving the same problem.

Each of these benefits can be obtained by using any of a host of different computer or programming systems. One such system which differs substantially from most of the rest and which has some very appealing features is a programmable desk-top calculator with graphical output. The particular system used for all the work described in this paper consists of a Hewlett-Packard 9100A calculator with a 9210A electrostatic printer and a 9125A *XY* recorder. This last item is a conventional 11″ x 17″ flat bed analog *XY* recorder equipped with a pair of storage registers and digital-to-analog converters. Our system cost about $8000. One valuable addition to the system to facilitate extensive student use would be the H-P 9160A optical card reader.

We have used this system to enrich the discussion of particle orbits in a power-law central-force field both with freshmen in an elementary physics course and with juniors in an intermediate mechanics course. In both cases the primary mode of use was as a lecture demonstration in which graphical solutions to the differential equations were generated for a variety of powers and initial conditions.

Figure 1 shows orbits obtained for inverse-square, inverse-fifth-power, and direct-twentieth-power force fields. The inverse-fifth-power case which produces a circular orbit passing *through* the force center (marked by a cross) especially amuses and intrigues students, prompting such questions as, "Does it really move like that? Wow!" (This particular question then led to a very

Department of Physics, Harvey Mudd College, Claremont, California.

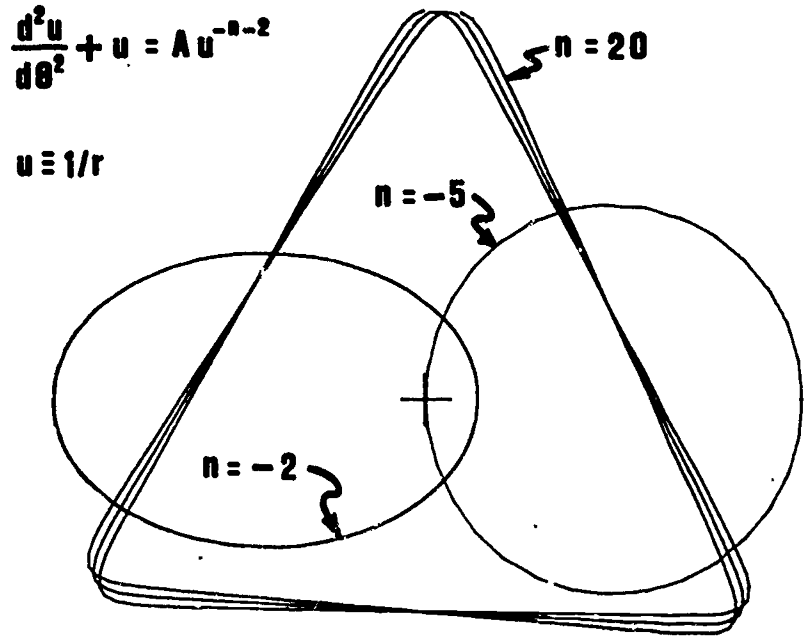


FIGURE 1 Sample orbits plotted on desk-top calculator system.

interesting discussion of the usefulness of a physical model with a non-physical value for one of its parameters.) The direct twentieth power is a useful approximation to a flat-bottomed potential well and, for these initial conditions, gave an interesting non-closed orbit.

One can study scattering trajectories using the same programs. Figure 2 shows two families of trajectories. The trajectories at the top represent r^{-2} force scattering of particles with equal initial velocities and various impact parameters (Rutherford scattering). The curves below represent scattering of particles with equal initial velocities and various impact parameters by a repulsive r^{-12} central force field. This is a crude model for the scattering of neutral gas atoms off one another at moderately high velocities. The circle shows that the neutral gas atoms behave nearly like hard spheres, whereas bare charges (the Rutherford case) experience a much softer barrier with the distance of closest approach much more dependent upon the scattering angle.

The main reason for using these programs was to enrich the student's intuition in this very important area of physics. Judging from their questions and excitement, we were successful. For the freshmen we had a second goal, to clarify for them what one means by a solution to a differential equation (and in particular to an initial value problem) by exhibiting a suitable numerical technique for solving such problems and then showing several solutions.

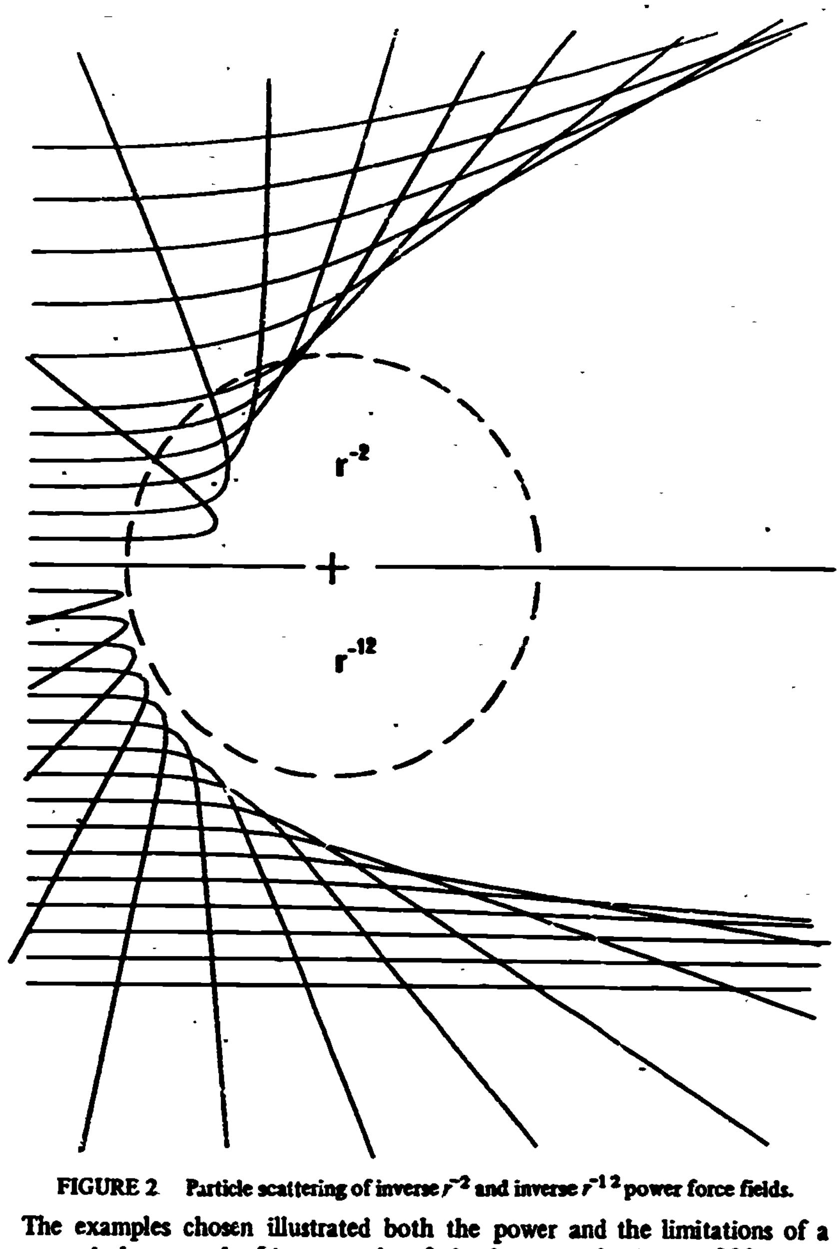


FIGURE 2 Particle scattering of inverse r^{-2} and inverse r^{-12} power force fields.

The examples chosen illustrated both the power and the limitations of a numerical approach. [An example of the latter is the inverse-fifth-power central force for which the radius goes to zero and the velocity to infinity at one point on the orbit. If one is integrating for the inverse radius $u(\theta) = 1/r(\theta)$ all goes well until the value of u exceeds the machine's capacity, at which point the pen stops essentially on the force center. If, on the other hand, one is integrating $\mathbf{F} = m\mathbf{a}$ directly with a constant time step the velocity of the pen will be proportional to that of the particle. As it approaches the force center the pen's steps will, therefore, become larger and larger until the truncation errors become enormous.] We believe that even this small experience with numerical solutions improved significantly our freshmen's understanding of differential equations and thus made the analytical solutions more readily acceptable to them.

The programs one writes for this sort of calculator/computer are quite different from those written for other computers. No computer language is needed. Instead one implements his algorithm directly as if at a super desk calculator. Perhaps the principal difficulty with this approach is that one must allocate storage oneself (there is no compiler that can process abstract symbols). Debugging a program on this type of machine is best done by executing it one step at a time and watching the variable values change.

Figure 3 shows a listing of the instructions stored for the program that generated Figure 1. Each step is equivalent to pushing some one of the 62 keys on the keyboard of the 9100A. The instruction code is a bi-octal number assigned to that key and is what the machine displays or prints when listing the program. The step number is the instruction's storage address. Fourteen instructions fit in each register. The program shown in Figure 3 assumes that one has loaded the force constant, A, and power, n, into registers a and b and also the initial values of θ, u, and $du/d\theta$ into registers e, d, and c, respectively. Register f is loaded with the (constant) step size, $\Delta\theta$. The program plots a point after every N calculations ($\Delta\theta$ steps) where the value $N-1$ is stored in the program in steps 3.1 to 3.3. If $A = 1$, a circular orbit will result for any value of n if $u = 1$ and $du/d\theta = 0$ at the start. The radius of the circle is inches times 500 is stores at locations 2.3 to 2.6. The main calculation loop is contained in steps 3.6 to 5.4. The numerical integration proceeds by the modified Euler method. Steps 5.5 to 6.2 determined whether or not to plot; the plot calculation is in steps 1.b to 3.5. The rest of the program converts the value stored in register b from n to $-n-2$ and changes the value stores in register c from $u'(\theta)$ to $u'(\theta + \Delta\theta/2)$ at the start and reverses both these changes whenever the program is stopped. To stop, one should press and hold the PAUSE key until the X, Y and Z registers light up, then press SET FLAG and CONTINUE. The parameters may then be modified and the program restarted by pressing CONTINUE. The three curves in Figure 1 were plotted in a little more than six minutes.

We have written two other programs for power-law central-force orbits, one in rectangular coordinates using a constant time step (for demonstrating Kepler's second law, etc.) the other incorporating general relativistic effects as a perturbation. Copies of these will be supplied upon request.

STEP	INST.	CODE	STEP	INST.	CODE	STEP	INST.	CODE	STEP	INST.	CODE
0.0	b	14	2.3	1	01	4.6	+	33	6.9	×	36
0.1	CHS	32	2.4	5	05	4.7	y→( )	40	6.a	a	13
0.2	↑	27	2.5	0	00	4.8	c	16	6.b	x⇄y	30
0.3	2	02	2.6	0	00	4.9	f	15	6.c	e^x	74
0.4	-	34	2.7	×	36	4.a	×	36	6.d	×	36
0.5	y→( )	40	2.8	÷	25	4.b	d	17	7.0	d	17
0.6	b	14	2.9	TO RECT.	66	4.c	+	33	7.1	-	34
0.7	d	17	2.a	FMT.	42	4.d	y→( )	40	7.2	f	15
0.8	ln x	65	2.b	÷	25	5.0	d	17	7.3	×	36
0.9	×	36	2.c	IF FLAG	43	5.1	RCL	61	7.4	2	02
0.a	a	13	2.d	6	06	5.2	+	33	7.5	÷	35
0.b	x⇄y	30	3.0	3	03	5.3	y→( )	40	7.6	c	16
0.c	e^x	74	3.1	0	00	5.4	e	12	7.7	+	33
0.d	×	36	3.2	0	00	5.5	1	01	7.8	y→( )	40
1.0	d	17	3.3	9	11	5.6	y⇄( )	24	7.9	c	16
1.1	-	34	3.4	x→( )	23	5.7	9	11	7.a	b	14
1.2	f	15	3.5	9	11	5.8	IF x>y	53	7.b	CHS	32
1.3	×	36	3.6	b	14	5.9	1	01	7.c	↑	27
1.4	2	02	3.7	↑	27	5.a	b	14	7.d	2	02
1.5	÷	35	3.8	d	17	5.b	-	34	8.0	-	34
1.6	c	16	3.9	ln x	65	5.c	y→( )	40	8.1	y→( )	40
1.7	x⇄y	30	3.a	×	36	5.d	9	11	8.2	b	14
1.8	-	34	3.b	a	13	6.0	GO TO	44	8.3	Cℓ x	37
1.9	y→( )	40	3.c	x⇄y	30	6.1	3	03	8.4	↑	27
1.a	c	16	3.d	e^x	74	6.2	6	06	8.5	↑	27
1.b	e	12	4.0	×	36	6.3	FMT.	42	8.6	STOP	41
1.c	↑	27	4.1	d	17	6.4	↑	27	8.7	GO TO	44
1.d	1	01	4.2	-	34	6.5	b	14	8.8	0	00
2.0	↑	27	4.3	f	15	6.6	↑	27	8.9	0	00
2.1	d	17	4.4	×	36	6.7	d	17	8.a	END	46
2.2	÷	35	4.5	c	16	6.8	ln x	65			

FIGURE 3 Program used to generate Figure 1. See text for instructions for use.

Our conclusion from using the H-P 9100 system for over a year is that introducing it as an in-class computer for demonstrations can be both easy and very helpful. One can, we are sure, devise many additional teaching uses for the machine including assigning homework problems to be done on it. (Before we do so, however, we would want to get the 9160A card reader to increase the speed with which students could enter their programs.) Without any formal instruction in its use or any assignments to be done on it, our one machine is already very heavily used by the students who often prefer it to our IBM 1620 (located in the same room) or to submitting programs to the local computer center's IBM 360/40.

Computation and Displays in Advanced Electrostatics with APL

LAURIE GAUVIN

INTRODUCTION

This paper describes a modest effort to take advantage of a time-shared IBM 360/50 computer for the purpose of demonstrating mathematical solutions to physical problems. The primary intention was to supplement the usual exercises by allowing some of them to be carried to completion, in the sense of evaluation, rather than being left at the stage of a formal expression. It should be recognized that this is a matter of some importance in the sciences since much of physics as it is actually taught and understood has to do with replacement of formal expressions that cannot be evaluated by others that can. The case in point consisted of replacement of the simple Coulomb law by the multipole expansion, which only required a very modest effort, using standard equipment and the APL language.

Regarding the mode of display, it is clearly necessary to produce numbers in many instances, particularly when concerned with accuracy. But one has to scrutinize a table of numbers very carefully in order to recognize irregularities or structure that would be quite apparent in the corresponding curves or to grasp the similarities that would be apparent in the plot of a family of curves. It becomes quite feasible to produce such graphs on a typewriter terminal provided that backspace and line feed movements are used to make the plotting speed acceptable. Such displays, in spite of their low resolution can provide quite adequate visualization of most functions of interest.

The course taught was one in advanced electromagnetic theory, leading to the solution of radiation fields from localized sources in the multipole expansion. Although it was a graduate course. 20 of the 23 students enrolled were seniors. The subject is treated using Green's function techniques which are introduced and exemplified in electrostatics. Thus the electrostatic multipole expansions are of interest because of their similarity to expansions of the radiation fields. It is particularly important that the student should grasp

Department of Physics, Laval University, Quebec, Canada.

their nature and behavior since the latter, when computed, can hardly be displayed effectively.

An APL program, called SPHERES, was written for the case of known axially symmetric potentials on two concentric spheres with free space elsewhere. The potential is expanded in spherical harmonic solutions of Laplace's equation,[1] and to evaluate it the program computes the coefficients of the Legendre expansion of the boundary potentials. The assigned problems had simple solutions involving factorials, but since approximately half the class had no prior contact with the APL system, programs for these were supplied. However, computed values can be displayed and checked. The main program computed radial or angular distributions at preset intervals and for one or more values of the other coordinate. Multipoles of order up to $l = 30$ can be included in typical computations without exceeding the available memory space.

Aside from the spherical harmonic expansion itself, the foregoing is essentially all the information that was provided the students regarding use of the programs. Some further discussion was given of a matrix formulation of the problem appropriate to APL. This was made quite straightforward by avoiding explicit arguments associated with function names, all necessary information being solicited by a printed request. Along with the programs were stored a description of the set and a simulated example. The former probably served the beginner mainly by supplying the identification of function names; the latter demonstrated the procedures rather effectively by using a slower typing speed (achieved with delay "characters") to simulate keyboard entries.

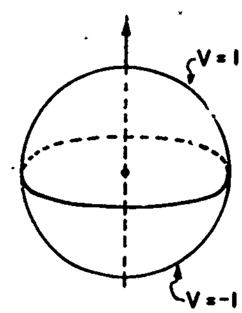
The students were instructed how to gain access to the computer, how to activate the stored programs and how to sign off properly. In addition, they were instructed to reactivate (load) the stored programs in case of abnormal program interruption. In short, essentially no instruction. The APL library contains a programmed learning course, and an instruction booklet[2,3] was available at the bookstore, and some students availed themselves of this opportunity.

GRAPHIC DISPLAY

A major problem with regard to the feasibility of producing graphs on a typewriter terminal is that of plotting speed. One solution is to move directly from one plotted point to the next by using backspace and line feed. In APL this presented problems: computer codes for these movements exist, but they cannot be generated from the keyboard and must be acquired initially with the assistance of system programmers. Once acquired, however, they behave as ordinary literal characters and can be manipulated as such. Credit for solving this problem is due to Pierre Laverdière and Denis Samson, students in physics at Laval University.

The typewriter (IBM 2741 Terminal) is rather prone to malfunction on execution of these movements, but this can be effectively overcome in the interest of reliability by insertion of special delay "characters" preceding the first backspace of a sequence and following line feed. The problem with line

```
      GRAPHIQUE
FN:
⎕:
      SPHERES
RAYONS DE POT. CONNU: A+B=?
⎕:
      0
COEFF. EN A:
⎕:
      ECHELON
L MAX:
⎕:
      31
1:DIST. RADIALE      2:ANGULAIRE
⎕:
      2
RAYON: R+A=?
⎕:
      .5 .9 1 1.05
```




```
L MAX=31 16TERMES≠0

+\⋆R=0.5  0.9  1  1.05
Δθ=5°   ANTISYMETRIQUE PAR RAPPORT A PI÷2
* R=.5
• R=.9
. R=1
+ R=1.05
```

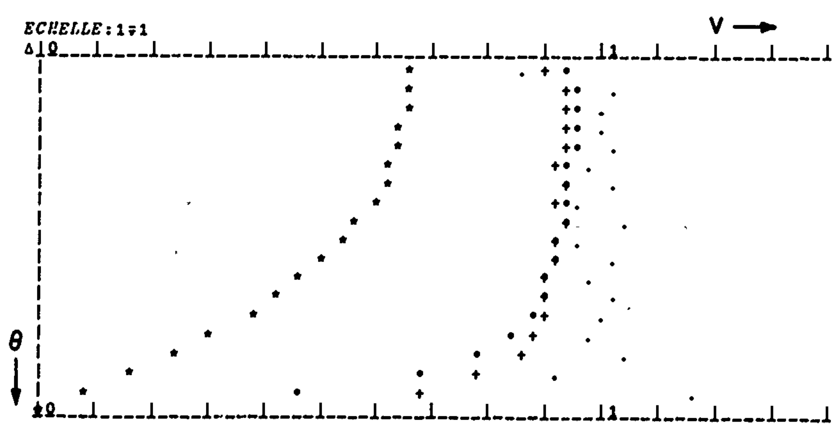


FIGURE 1 A complete computation. Keyboard entries are indented except for the plot characters and legend entered just before the graph. The potential is plotted between 0° and 90° in steps of 5°.

feed is that it sometimes shifts to uppercase. The left margin is sometimes mechanically indefinite and may cause a portion of a curve to be shifted, but this occurs very infrequently.

A set of curves is represented as a matrix of ordinate values corresponding to a set of equally-spaced abscissas, each row-vector or column-vector representing a curve. A plotting program was written which converts such a matrix into a graphic display on the typewriter terminal, proceeding along columns or rows of the matrix according to which are the longer (it is assumed that there are more points on each curve than there are curves to be plotted). Ordinate scale markings are produced, and an optimum scale is selected so that scale units are round numbers. The rounded range corresponds to 50 print positions; the actual graph may have a range somewhat more or less than this (maximum approximately 70).

The plotting speed achieved is psychologically quite acceptable. In a typical case such as Figure 1 each curve is plotted in approximately 20 seconds and the complete graph in about 2 minutes, a good deal faster than printing the corresponding table of numbers. The students did not display numbers except for special purposes, and indeed this was quite proper. Much of physics has been discovered by graphically displaying significant features of reality. Thus it was suggested to the students that they might consider their assignment as one of measurement of the potential at appropriate points and of plotting their results in the form most suitable for comprehension. From this point of view their task was to discover the regions where the potential presents significant structure and to select the curves accordingly. Also since a fixed number of terms is included for all points entering a given computation, insufficient convergence of the series will usually become apparent for points that lie on or very near a physical boundary. The very rapid damping out of corresponding spurious oscillations in the angular distributions as one moves away from the boundary is easily observed.

Examples of typical computations are given in Figures 1 and 2. In Figure 1 ECHELON is the name of the programmed formula for coefficient of the Legendre expansion of the odd step function whose value corresponding to two oppositely charged halves of a shere is -1 for $-1 \leqslant x \leqslant 0$ and 1 for $0 \leqslant x \leqslant 1$. Following the header which marks the end of the computation by SPHERES there follows a request by GRAPHIQUE for plot characters. This is signaled by one or more line feeds, but no printout in order to keep the presentation clean. Characters beyond the first are ignored but may be used to type in a legend, as illustrated. At the bottom of Figures 1 and 2 a triangle and arrow are used to signal that the paper should be rolled back to the preceding triangle to plot the next curve, and there is delay for this purpose. These programs are much more compact than the corresponding FORTRAN programs would be; copies can be obtained by writing to the author.

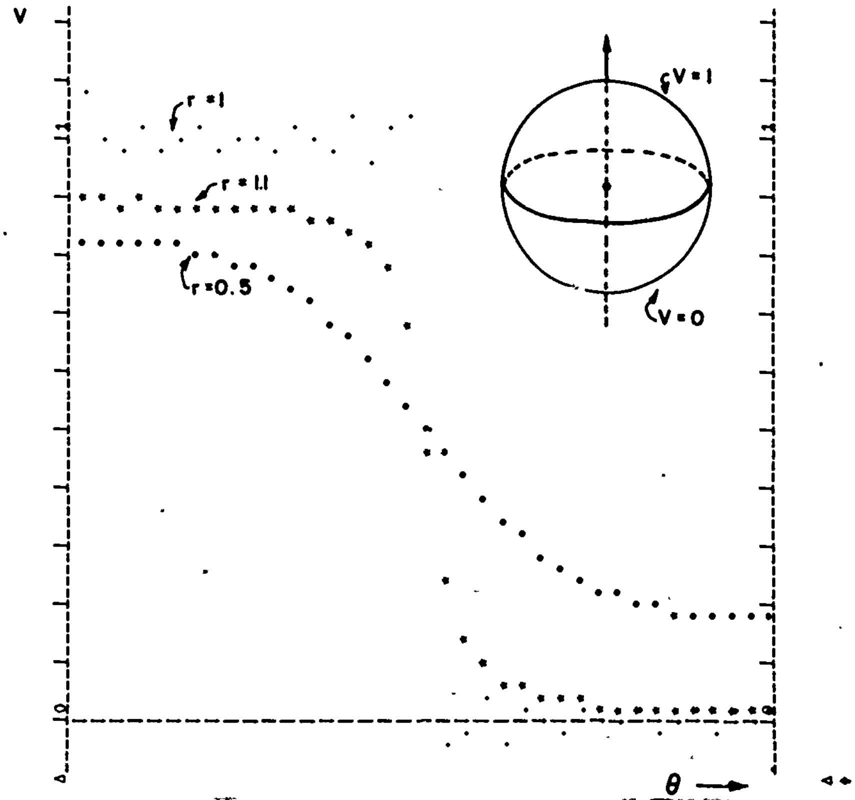


FIGURE 2 A typical graph. The potential is plotted between 0° and 180° in steps of 5°. Spurious oscillations due to truncation of the series may be seen on the curve for $r = 1$.

CONCLUSION

Such use of a computer requires very modest effort and cost; total computer bills were \$325, or about \$15 per student at \$3 per assigned problem. The cost of the initial programming might reasonably be assessed at \$1000. Students worked at the terminals an average of 7 hours, so it is reasonable and sensible for a teacher to include such use as a supplement to the usual exercises. It might be useful to suggest a comparison with the use of oscilloscopes in the laboratory. The costs hardly reflect the superior power and accuracy of the computer.

REFERENCES

1. J. D. Jackson, *Classical Electrodynamics*, Wiley, New York, 1962, p. 67.
2. A. D. Falkoff and K. E. Iverson, *APL/360 User's Manual*, IBM, 1968.
3. K. E. Iverson, *A Programming Language*, Wiley, New York, 1962.

Advanced Data Analysis for Undergraduates

R. A. BURNSTEIN, D. L. SWANSON, and V. R. VEIRS

Since computing facilities are available on most campuses, it is appropriate and, indeed, desirable to encourage the students to gain experience in the use of sophisticated data analysis techniques and in the use of the computer. A FORTRAN computer program[1] (DATANL) for the senior-level physics laboratory has been developed at the Illinois Institute of Technology (IIT) which allows students to use advanced data analysis techniques not normally available to them. The program is divided into three sections: Section A determines the best values of measured data by applying statistical methods to the data taken by the student; Section B computes desired quantities as a function of the data and estimates the errors in them from the random errors in the measured data; Section C may be used to fit various distribution functions to the data of A or the computed quantities of B.

Data that are assembled by students in the laboratory are usually obtained in one of two formats; either a sequence of repeated measurements of several variables is made at each of one or more points, or a single measurement of each variable is made at each of the serveral different points. Ordinarily, an estimated error accompanies each measured variable. In the case of repeated measurements, the first part of the computer program (A) is used to reduce the measured data to a set of mean values, with their errors, corresponding to each set of measurements. This is achieved by calculating[2] the means and standard deviations or the weighted means and weighted standard deviations when estimated measurement errors have been recorded. The repeated measurements are then examined for consistency by using chi-square[3] calculations to obtain the probability that the measurements are randomly distributed about their means and also to check whether the measurement errors have been assigned properly. At this stage, data points that fall outside[4] the expected distribution may be deleted if desired. If this is done, new means and standard deviations are calculated for the remaining sample of measurements. After this analysis has been performed, a summary is printed out

Department of Physics, Illinois Institute of Technology, Chicago, Illinois.

TABLE 1. Measurements of Impact Parameter vs. Arc Lengths of Scattered Particles

Impact Parameter (cm)	Arc Lengths (cm) (from combined readout of repeated measurements) (data recorded in order of increasing length)									
Right Side										
0.141	1.35	1.85	2.25	2.35	2.35	2.4	2.45	2.55	2.6	2.6
0.282	4.5	4.75	4.85	4.9	4.95	5.15	5.2	5.65	4.8	4.9
0.423	6.4	6.9	7.05	7.35	7.75	7.8	8.05	8.1	8.3	8.2
0.564	9.6	9.9	9.95	10.05	10.05	10.1	10.2	10.3	10.35	10.95
0.705	11.85	12.75	12.95	12.95	12.9	13.1	13.15	13.7	13.8	14.3
0.846	14.85	15.25	15.3	15.55	15.55	15.8	15.95	16.0	16.1	16.3
0.987	18.55	18.7	19.0	19.3	19.35	19.45	19.6	19.95	19.9	20.2
1.128	21.1	21.9	22.15	22.15	22.35	22.9	22.95	22.95	23.5	24.05 All
1.269	24.6	25.2	25.5	25.55	25.95	26.05	26.15	26.25	26.25	26.95 pts.
1.410	28.35	28.5	28.75	28.85	28.85	30.0	29.3	29.4	29.6	29.9 ±0.1
Left Side										
0.141	2.1	2.15	2.3	2.45	2.45	2.35	2.6	2.65	2.8	3.05
0.282	4.45	4.65	4.75	5.0	5.0	5.1	4.8	5.4	5.45	5.75
0.423	7.25	7.35	7.35	7.55	7.55	7.60	7.8	8.05	8.25	8.7
0.564	9.35	9.45	9.65	9.7	9.85	9.85	10.0	10.15	10.65	10.7
0.705	12.2	12.4	12.3	12.65	12.65	12.65	12.7	12.7	12.85	12.9
0.846	14.75	15.25	15.3	15.3	15.3	15.4	15.45	15.8	16.2	16.2
0.987	17.4	17.5	18.0	18.15	18.4	18.5	18.8	18.85	19.05	19.2
1.128	20.1	21.0	21.05	21.05	71.2	21.35	21.4	21.65	21.85	22.5
1.269	23.95	24.35	24.55	24.7	22.05	25.25	25.5	25.55	25.8	25.9
1.410	27.35	27.55	28.05	28.4	28.55	28.55	28.65	28.65	28.85	29.15

which includes a list of raw measurements, a list of any deleted points, the calculated means and standard deviations, the probability that the measurements are consistent and a histogram of the repeated measurements.

As an example of a simple application of part A of the program to an actual experiment, we show in Table 1 data compiled from a scattering experiment.[5] Small steel ball bearings were fired at a cylindrical target, and the scattering angle was measured as a function of the impact parameter. Repeated measurements of the scattering angle (simply related to the arc lengths) were made at each of several different impact parameters. These numbers were punched onto computer cards and analyzed by part A of the computer program. The results are shown in Table 2. The mean arc length and its statistical error, *i.e.*, standard deviation, have been calculated as a function of the impact parameter. In addition, the computer program plots a histogram, shown in Figure 1, of the deviations from the mean for all the measurements. The smooth curve superimposed upon the histogram is a Gaussian distribution which was fitted to the measured deviations in order to show that the measurements were drawn from a random sample and are not strongly influenced by systematic effects.

In part B of the program, quantities may be calculated from the measured data by utilizing a specific functional relation, introduced into the program by the user, in the form of a FORTRAN function subroutine. The function is evaluated at each of the best values of the measured data (calculated in part

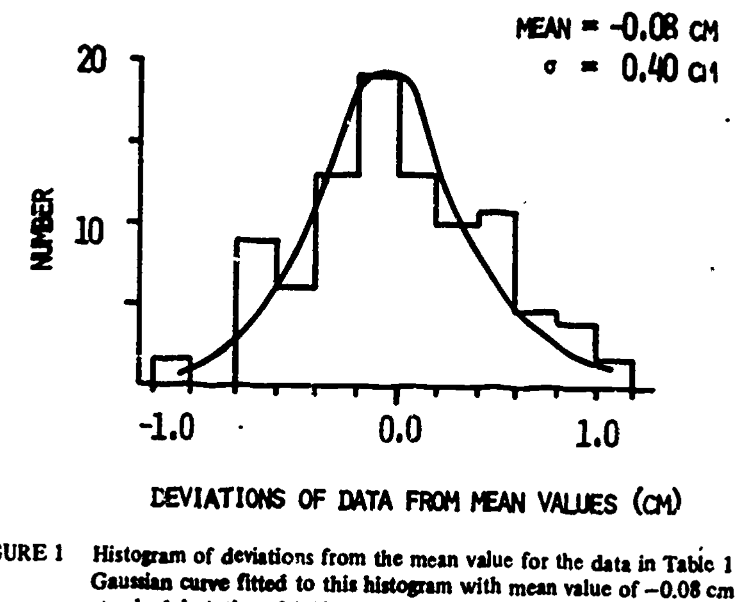


FIGURE 1 Histogram of deviations from the mean value for the data in Table 1 and Gaussian curve fitted to this histogram with mean value of -0.08 cm and standard deviation of 0.40 cm.

TABLE 2 Determination of Mean Value and Standard Deviation for Data in Table 1

Impact Parameter (cm)	Arc Length (cm)
0.141	2.415 ± 0.579
0.282	4.939 ± 0.299
0.423	7.522 ± 0.601
0.564	10.15 ± 0.321
0.705	13.14 ± 0.636
0.846	15.63 ± 0.432
0.987	19.40 ± 0.512
1.128	22.54 ± 0.792
1.269	25.84 ± 0.622
1.410	29.15 ± 0.545

A), and the errors in the calculated quantities are determined numerically by propagating the statistical errors (or the estimated measurement errors if repeated measurements have not been made) through the given function. The entire error matrix is calculated and is used to determine the correlation between the errors in a measured variable. Table 3 shows the results of

TABLE 3 Radius Determination at Each Value of the Impact Parameter

Impact Parameter (cm)	Target Radius (cm)
0.141	3.40 ± 0.92
0.282	3.33 ± 0.23
0.423	3.28 ± 0.30
0.564	3.25 ± 0.12
0.705	3.14 ± 0.17
0.846	3.18 ± 0.09
0.987	3.00 ± 0.09
1.128	2.97 ± 0.11
1.269	2.94 ± 0.08
1.410	2.92 ± 0.05

the application of part B of our program to the scattering experiment. From the data presented earlier, the radius of the scattering target has been calculated for each value of the impact parameter. The calculated radii and their numerically determined errors are shown in the right-hand column.

If there are unknown parameters in the functional relation that is thought to represent the data, then part C of the computer program may be used to determine the best values and the errors associated with these parameters. Fits may be performed to either the best values of the measured data from A or the quantities calculated in part B. The best fit is obtained by utilizing the maximum likelihood[6-8] technique which leads to a search for the minimum of a chi-square (χ^2) function. If the desired distribution function is linear in the parameters, then the best values of the parameters are found by calculating the least-squares solution, solving for the unknown parameters in a set of simultaneous linear equations. If the desired distribution function is non-linear in the parameters (which is often the case), then the best fit is obtained by searching for the minimum in a multidimensional χ^2 space.[9]

Some widely used distribution functions such as power series, Legendre polynomials, Gaussian, and Poisson distributions have been built into the program. In addition, any specific function desired by the user may be included simply by writing an appropriate FORTRAN function subroutine. The fitting procedure calculates the best values of the desired parameters and also calculates the errors in these parameters. Furthermore, it prints out all the linear correlation coefficients which are a quantitative measure of the degree of correlation between all pairs of fitted parameters. Also printed out is the

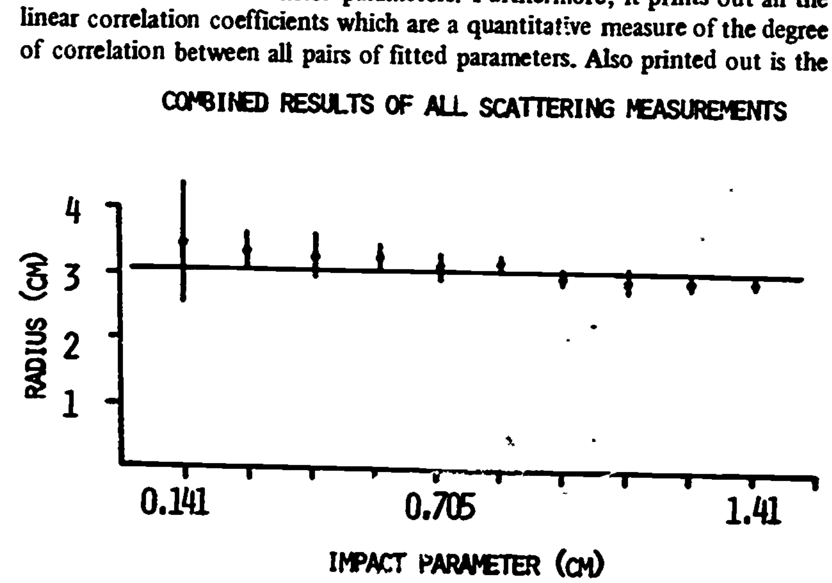


FIGURE 2 A plot of the value of the target radius vs. impact parameter and the best-fit value determined for the radius (3.01±0.03 cm).

minimum chi-square and the probability that a randomly selected experiment would give this value of chi-square.

Some of the results of applying part C of the program to our simple scattering experiment are presented in Figure 2. Ten values of the target radius have been calculated in part B and are fitted to the constant term in a power series. The line drawn is the best fit to the radius, (3.01 ± 0.03) cm. The minimum χ^2 (=13.4) gives a probability of 18% that these measurements are consistent.

Figure 3 shows a flow chart of the computer program. The operational aspects of the program have been designed so that it is easy to prepare the input data and utilize various program controls which govern the statistical procedures, calculations and fits. In part A data are read in and statistical quantities are calculated. If desired, the data can be modified by eliminating data points what are far from the expected distribution, and the statistical calculations are performed once again. In part B a given function is introduced and evaluated. When appropriate, χ^2 can be calculated to
tively compare the function to the data. In part C variable parameter fits are

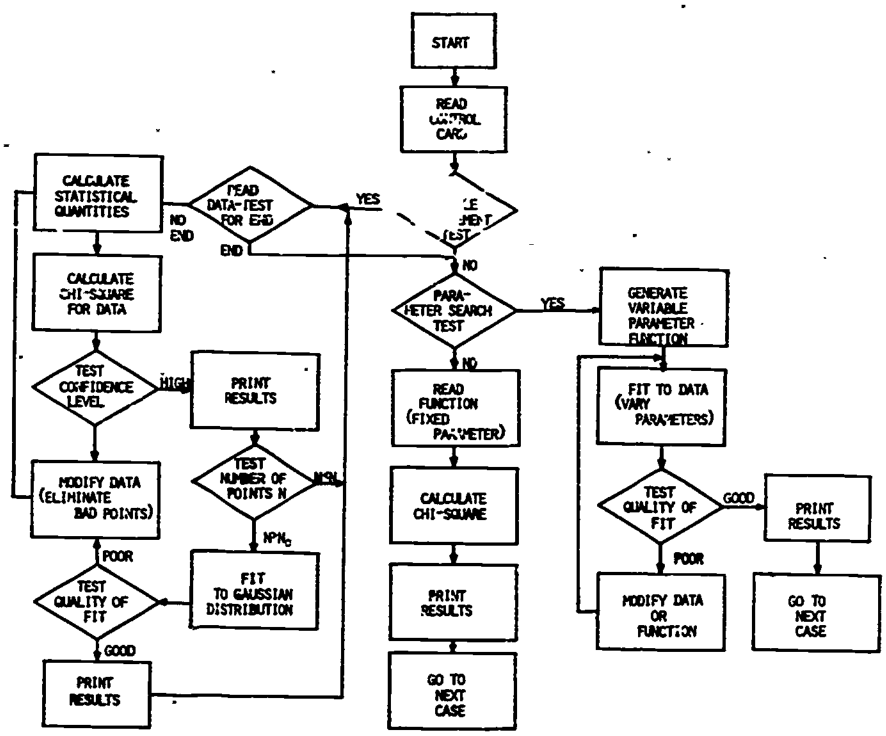


FIGURE 3 Flow chart of the program DATANL.

performed in order to obtain the best values and the errors of the parameters associated with a desired distribution function.

In summary, we have developed this data analysis computer program for the use of IIT's senior undergraduate physics students with several thoughts in mind. Students can enlarge the scope of their laboratory work by taking many repeated measurements, which aids in the understanding of both measurement errors and their effect on the uncertainty of a physical result. Using the computer, physical conclusions can be drawn from the data by utilizing computationally complex analysis techniques which enable the student to determine not only the best values of parameters associated with his experiment, but which also give quantitative statements about the errors associated with the parameters and the correlations between the parameters. This ensemble of fitted results, errors and correlations allows a most accurate statement of physical conclusions. Finally, as a by-product, the student learns enough FORTRAN programming to adapt our program to his experiment.

REFERENCES

1. A listing of the program is available upon request from the authors.
2. For the computation of statistical quantities, we follow the formulas given by standard treatments of statistics. See, for example, B. R. Martin, "*Basic Statistics for Physicists*, Brookhaven National Laboratory internal report BNL-12602, 1968, and references contained therein.
3. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1965.
4. We set three standard deviations as our rejection level. However, this criterion is an adjustable parameter of the program.
5. This example has been chosen to display the use of the various parts of our program. We are aware that there may be other ways of illustrating the physics of scattering.
6. J. Orear, *Notes on Statistics for Physicists*, University of California Report UCRL-8417, 1958.
7. P. Bevington, *Data Reduction and Error Analysis for the Physiccl Sciences*, McGraw-Hill, New York, 1969.
8. D. C. Baird, *Experimentation, An Introduction to Measurement Theory and Experimental Design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.
9. We use the general minimizing computer program MINIMZ developed at the University of Wisconsin by E. West and J. Boyd (private ccommunication).

The Mechanization of Mathematics

CARL ENGELMAN

INTRODUCTION: THE NEED

One possible first reaction to the intentionally provocative title of this paper is that mathematics, unlike the physical sciences, has no technology, at least no "hard" technology. But it does; a technology so simple as to be almost invisible. It is the technology of the scratchpad, the blackboard, the printed page; a technology that has changed very little since Archimedes scratched his figures in the sands of ancient Syracuse. Yet, it is a technology that stands today on the threshold of a revolution that will see the scratchpad replaced by the cathode-ray-tube and the textbook by the secondary storage of large-scale computer systems.

The "soft" technology of mathematics, that of symbology and of conceptual tools, will be augmented by extensive programs which will provide on-line aid in all the mechanical processes of applied mathematics—substitution, simplification, differentiation, integration, series expansion, integral transforms, matrix inversions, etc. These procedures will all be available to the mathematician on a full-time basis, and they will all operate either symbolically or numerically according to his wishes. Many of these programs will be based on algorithms and techniques outside the comprehension of their users. Engineers will integrate symbolically like most of us operate television sets today. Of course, engineers do this now—just so long as the answer is in a published table of integrals. In the future the machine will compute the answer, and it will have algorithmic methods that will take it far beyond the scope of any contemporary tables.

We anticipate, then, that each mathematician will have a mechanical assistant possessing an encyclopedic knowledge of mathematical formulae and results, capable of rapid and flawless computation, familiar with a broad spectrum of computational algorithms, waiting silently by his desk twenty-

The MITRE Corporation, Bedford, Massachusetts. Development of the MATHLAB program reported herein was sponsored by The MITRE Corporation Independent Research Committee. It was also supported, in part, through access to computer facilities, by Project MAC, an M.I.T. Research Program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number NONR-4102(01).

description of what the practice of mathematics will be like in the not distant future, we must add that there are more compelling reasons for the implementation of interactive mathematical laboratories.

The most obvious such reason is the inhuman demands posed by really enormous computations such as those associated with the many-body problems of analytic dynamics. A few such computations, e.g., Delaunay's calculation of the motion of the moon under the gravitational attraction of both the sun and the earth, have been completed by hand.[1] But most such computations are abandoned as just too tiring (they literally take years) and too error-prone to justify the human sacrifice involved. In the case of Delaunay, the computation took twenty years and filled some 800 printed pages. Almost as we are writing this, it has been announced that, for the first time, this computation has been repeated (in fact, extended) symbolically by machine.[2] Delaunay did err, but what is astounding is how far one has to proceed in the computation before any error does occur.

When we turn to what is, in our minds at least, the most important justification for developing such aids, we are ironically helpless to predict the consequences. The goal we have in mind is the liberation of the mathematician from drudgery. It must be true that an applied mathematician (and within the scope of this phrase we include, of course, physicists, engineers, astronomers, anyone who is engaged in the *practice* of applied mathematics) must spend most, quite possibly eighty to ninety percent, of his time in purely mechanical computation. Relatively brief intervals in which he is making decisions as to the next few steps are interspersed with long ones in which he is functioning as a rather inefficient computer, performing the work of drones and then, with even less relish, checking and rechecking the accuracy of these computations. What would it mean if one had available to perform these computational tasks, as well as serving as secretary and librarian, easy access to the sort of system we envision? The computer would act as a mechanical amplifier which would allow the mathematician to almost flit from idea to idea, painlessly testing the consequences of each and with no undue investment in any of them. It is a power which, quite frankly, might be resented by those accustomed to less frequent demands being placed upon their thinking. But its consequences for young mathematicians are beyond our imagination.

INTERACTIVE SYMBOLIC COMPUTATION

For the most part, we are concerned with symbolic rather than numerical assistance for the mathematician. That is, we are much more concerned with the manipulation of mathematical expressions than with their evaluation. This emphasis should not, in any way, be taken as an attempt to denigrate the importance of numerical aids for mathematicians. In fact, what is certainly needed is an integrated system capable of both the symbolic computations that derive the relevant expressions as well as the numerical computations associated with the evaluation or solution of these expressions when numerical values are specified for the symbolic parameters that appear in them. The emphasis on symbolic computation is due to its being much less understood at this point, so that the critical problems that will decide whether such an integrated system succeeds or fails will most likely be associated with its symbolic facilities.

Symbolic computations place much heavier burdens on the interactive aspects of the system. Certainly a physicist or engineer would prefer an on-line presentation of his solution space which would graphically demonstrate the consequences of his varying the parameters available to him. Such facilities should be contained within our prospective system. Compared, though, to the situation that prevails as soon as we consider symbolic computation, we are tempted to characterize these graphical display programs as existing at a qualitatively lower level of interaction. The success of symbolic computation is generally dependent on a much more intimate, continuous and complex symbiosis than is the case when the computations are purely numerical. The reason for this is that, at any step in the solution of a symbolic problem, we are dealing with expressions that may be expressed in thousands of mathematically equivalent forms. The choice of which of these thousands is the one we actually employ will affect not only the transparency of the answers but might also determine whether the intermediate expressions explode to the point of destroying the computation.

We shall cite our own program as an example, it being unnecessary to embarrass others.[3,4] MATHLAB 68, if presented with the expression

$$\mathrm{ARCTAN}\left(\frac{X+A}{1-X*A}\right) \qquad (1)$$

and asked to differentiate it with respect to X, would yield the answer

$$\frac{\dfrac{A(X+A)}{(1-X*A)^{2}} + \dfrac{1}{1-X*A}}{1+\dfrac{(X+A)^{2}}{(1-X*A)^{2}}} \qquad (2)$$

Now this answer may be reduced to:

$$\frac{1}{X^2+1} \tag{3}$$

While MATHLAB is capable of this reduction, it must be guided to it by the user. It would suffice, for example, for the user to type:

'RATSIMP ('WS)$

This is an abbreviation for "RATional SIMPlification of WorkSpace"; "Workspace" referring to the most recently computed result. The significance of the single quote marks will be explained later. Needless to say, if the expression (2) were a partial result and the user was not there to intervene early in the game, this monstrosity would propagate swiftly, destroying the computation like the dominant lethal mutation that it is.

Symbolic expressions, then, represent a greater challenge for us not only because they are a more novel, complex or abstract species of data, but also because they place much greater demands on the interactive facilities of the system. However, it would not be accurate to attribute the need for interaction entirely to the differences between numerical and symbolic computations. What matters, of course, is how dynamic the situation is; what chance one has of interrupting the process and influencing its future. A rather beautiful example of how numerical programs might demand on-line facilities is presented by the programs of Fromm[5] for the display of solutions of non-linear flow problems. The future mechanized mathematical laboratory must certainly provide the capability to view such "movies" on the face of the same cathode-ray-tube on which the differential equations are manipulated symbolically.

INTIMATIONS OF THE FUTURE: MATHLAB

Lest our predictions of the future of applied mathematics be dismissed as pure bravado, we should like to cite one experimental program, our own, to demonstrate that many of our predictions have already been realized, at least *in vitro*. MATHLAB[3] is an on-line experimental system providing machine aid for the mechanical symbolic processes encountered in analysis. It is capable of automatically and symbolically performing such common procedures as differentiation, polynomial factorization, indefinite intergration, direct and inverse Laplace transforms and the solution of linear differential equations with constant (symbolic) coefficients.

This is not to say that our system is the only one of interest. Readers interested in a survey of what has been accomplished by others are referred to Reference 4 at the end of this paper. Nor are we trying to give the impression that all the problems have been solved. They certainly have not, and that refers to both technical and economic problems. Outstanding among the technical problems are those centering upon such issues as the handling of very large mathematical expressions and the user control of the various simplification transformations. Nonetheless, it is our contention that there has already been sufficient success that one must observe the handwriting on the wall.

We should like to present two examples of MATHLAB in action. Each is chosen partially for its brevity, and, as a result, only a small portion of the actual facilities supplied by MATHLAB are exhibited. The first example is chosen to illustrate what might be considered the normal usage of MATHLAB, namely, an interactive mode in which the user guides the computation on a line-by-line basis. The second example is also interactive, but in this instance we have chosen an area in which MATHLAB possesses some expertise; and, consequently, we shall see that the computer dominates the problem-solving process.

Example 1: A Problem in Particle Mechanics[6]

We suppose a dynamical system with two degress of freedom and generalized coordinates q_1, q_2, p_1, p_2. New coordinates Q_1, Q_2, P_1, P_2 are introduced by the equations:

$$\begin{gathered} Q_1 = q_1{}^2 + \lambda^2 p_1{}^2; \quad Q_2 = q_2{}^2 + \lambda^2 p_2{}^2 \\ P_1 = \arctan (q_1/\lambda p_1) - \arctan (q_2/\lambda p_2); \quad P_2 = \lambda \arctan (q_2/\lambda p_2) \end{gathered} \tag{4}$$

We are to show that this is a contact transformation.

This is a purely formal mathematical problem which can be formulated[7] as follows: the Poisson bracket of two functions, U and V, of the variables q_1, q_2, p_1, p_2 is denoted by (U, V) and is defined as

$$(U, V) = \sum_{j=1}^{2} \left(\frac{\partial U}{\partial p_j} \frac{\partial V}{\partial p_j} - \frac{\partial U}{\partial p_j} \frac{\partial V}{\partial p_j} \right) \tag{5}$$

The conditions that the above transformation constitutes a contact transformation are

$$(Q_i, Q_j) = (P_i, P_j) = 0 \qquad (i, = 1, 2 \text{ and } j = 1, 2) \qquad (6)$$
$$(Q_i, P_j) = 0 \qquad i \neq j$$
$$(Q_i, P_i) = 1 \qquad i = 1, 2$$

Equation (5) reveals immediately that the Poisson bracket is antisymmetric; i.e., $(U,V) = -(V,U)$ and, in particular, that $(U,U) = 0$ for all U,V. Hence, the conditions (6) reduce to: $(Q_1, Q_2) = (P_1, P_2) = (Q_1, P_2) = (Q_2, P_1) = 0$ and $(Q_1, P_1) = (Q_2, P_2) = 1$. We shall attempt, with the aid of MATHLAB, to confirm these conditions.

The MATHLAB conversation that follows is composed of three species of text. The prose passages within double parentheses are explanatory comments. The number sign is MATHLAB's signal that it is listening and the dollar sign is the terminal character for MATHLAB input. So any text between a # and a $ may be assumed to have been typed by the user. Any line which is not input constitutes the computer's response to the most recent user request. Since we have neither subscripts nor lowercase alphabet, we shall refer to the old and new variables as Q1,...,P2 and BIGQ1,...,BIGP2, respectively. We now introduce the transformation by means of four assignment statements. BIGQ1: . . .$ means assign the expression to the right of the colon as the value of BIGQ1, etc.

```
#BIGQ1  :  Q1↑2+L↑2*P1↑2$

  2   2 2
Q1  + L P1

#BIGQ2  :  Q2↑2+L↑2*P2↑2$

  2   2 2
Q2  + L P2

#BIGP1 : ARCTAN (Q1/ (L*P1)) – ARCTAN (Q2/ (L*P2))$

            Q1                       Q2
ARCTAN ( ------ )  -  ARCTAN ( ------ )
           L*P1                     L*P2

#BIGP2 : L*ARCTAN (Q2/ (L*P2))$

            Q2
L*ARCTAN ( ------ )
           L*P2
```

The word for differentiation in MATHLAB is DERIV. In order to avoid typing it several times, we introduce D as an abbreviation for DERIV by typing in

```
#ALIAS DERIV D$
```

We now have to explain the most difficult point in connection with these examples—the user's control of evaluation. Should the following explanation seem too complicated, we urge the reader to scan the examples anyhow. It is possible to get an idea of the services MATHLAB might provide while trusting the author to get the quote-marks right.

Within MATHLAB, a variable is quoted, i.e., evaluates to its literal self, unless the user precedes it by a single quote-mark (') which we read as "unquote" or "eval." In this latter case, MATHLAB will substitute whatever *mathematical expression* has most recently been assigned as the value of that variable. This convention also refers to functions as well as variables. The input 'F(...) causes the function F to be evaluated, while the input F(...) is purely formal. So, for example, if Y is assigned the value SIN(X), e.g., by the previous assignment statement Y:SIN(X)$, then:

```
                                DY
DERIV (Y,X)     evaluates to   -----
                                DX

                                D
DERIV ('Y,X)    evaluates to   -----    SIN (X)
                                DX

'DERIV(Y,X)     evaluates to   0

'DERIV('Y,X)    evaluates to   COS (X)
```

We wish now to define the Poisson bracket of *U* and *V* as a function, POISSON (U,V). In order to do this we must make reference to certain derivatives. We wish to type these in preceded by single quote-marks so that MATHLAB will understand that we intend that the differentiations be performed, not just referred to. However, we need some device which allows us to type the single quote-marks but inhibits their action to the extent that they take effect when we apply the function POISSON, rather than at the moment we type in its definition. The inhibitory mechanism is the double quote-mark ("), which we read as "quote." To define POISSON, the user may type:

```
#POISSON(U,V) : "('D(U,Q1)*'D(V,P1)
-'D(U,P1)*'D(V,Q1)+'D(U,Q2)*'D(V,P2)
-'D(U,P2)*'D(V,Q2))$
WS(U,V)
:
   'DERIV(U,Q1)'DERIV(V,P1) - 'DERIV(U,P1)'DERIV(V,Q1)
+
   'DERIV(U,Q2)'DERIV(V,P2) - 'DERIV(U,P2)'DERIV(V,Q2)
```

The most recently computed expression in MATHLAB is called "workspace" and is automatically named WS. The above response is, partially, an effort on the part of MATHLAB to warn us that the current content of workspace is a function.

We may now proceed to first turn on a switch called "SIMP," which guarantees that all our answers are "simplified" and then evaluate each of the requisite Poisson brackets in succession.

```
#SIMP ON$

#'POISSON('BIGQ1,'BIGQ2)$
0
#'POISSON('BIGP1,'BIGP2)$
0
#'POISSON('BIGQ1,'BIGP2)$
0
#'POISSON('BIGQ2,'BIGP1)$


                                    2
          2L                     2Q2
  ----------------- +  ----------------
          2                        2
       Q2                 2     Q2
    -------- + 1        P2 L (------- + 1)
       2 2                        2 2
     P2 L                       P2 L
```

While this is "simplified," it isn't simple. We need RATSIMP to combine this into a single fraction, performing all possible division cancellations.

```
#'RATSIMP('WS)$
 2L
```

Note: the answer, according to Equations (6), should have been 0. MATHLAB's conclusion is, then, that the given transformation is *not* a contact transformation. The problem was changed considerably in a later edition[8]; when we tried that version with MATHLAB, we found the transformation correct, except for a minus sign.

Example 2: Third-Order Linear Differential Equation with Constant Symbolic Coefficients

In the previous example, we exhibited the expressions as they would be "displayed" two-dimensionally on a teletype. The display programs, in fact, require only the facility to print a given character at a given place. They have functioned on teletypes, line printers, IBM 2741 typewriter terminals, and alphanumeric CRT terminals as well as graphical display consoles and plotters. We shall present the second example as it appeared on the face of a graphical display (DEC 340) in making the MATHLAB 68 film exhibited at the Conference on Computers in Undergraduate Science Education. The differential equation is displayed in Figure 1. The conversation accompanying the solution is repeated in Figure 2. Note how the computer guides the conversation. About all that has to be explained is that, when we say ALLFORMAL, we abdicate the privilege of specifying the initial conditions. The machine chooses general constants which it calls Y(O), Y'(O), and Y''(O). The solution is displayed as Figure 3.

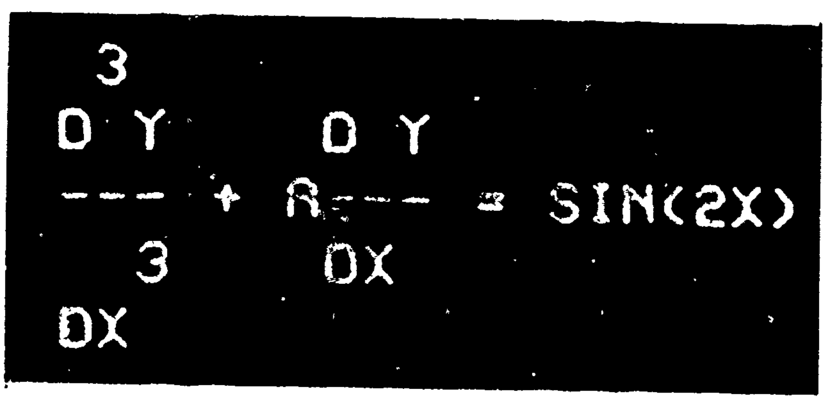


FIGURE 1   Example 2, the differential equation.

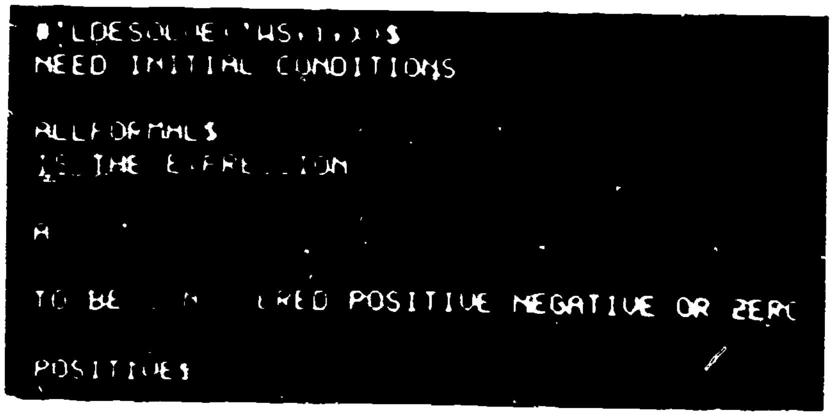


FIGURE 2 Example 2, conversation accompanying the solution.

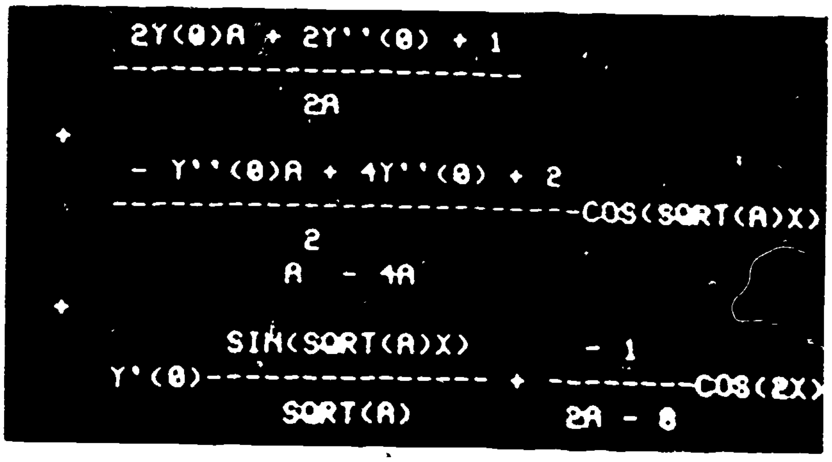


FIGURE 3 Example 2, the solution.

The most notable thing about this example is how much of the burden is borne by the machine. It is the machine that (1) decides to solve the equation as an initial value problem, (2) recognizes the equation as third order and so determines the necessity of specifying the value of the solution and its first two derivatives at the origin, (3) presses the user for these initial conditions, (4) chooses an appropriate method of solution (it in fact employs both direct and inverse Laplace transforms), (5) realizes the ambiguity contingent upon the sign of 'A' and again makes inquiry of the user, and (6) presents the answer in a format of its choosing. When the commands available to the user are so sophisticated, e.g., "solve this differential equation," then it becomes possible for the user to operate very close to a purely conceptual level.

Implications for Undergraduate Education

The MATHLAB program just discussed was designed as an experimental symbolic computational aid not as an instrument of instruction. Yet, we believe that it presents some interesting challenges in undergraduate education. Consider, for example, the program for symbolic differentiation. It would entail a fairly minor alteration to make it print out, during a differentiation exercise, each and every invocation of the chain rule. Furthermore, a teaching program for differentiation constructed from the routines internal to MATHLAB would have, as an advantage over more conventional CAI programs, the fact that it was based, not on its creator's hope that he could anticipate all the student's problems, but rather on the program's own understanding of the relevant algorithms. The program would have genuine knowledge; it would know how to differentiate any expression. This would mean, for example, that the student could query the program: "How do you differentiate this?" The program could respond by performing the requested differentiation, pausing at each step to explain what it was doing.

Another possible approach to computer usage in science education is predicated upon our possession of programs that understand the symbolic processes of calculus. An example would be a program to check, line by line, a student's attempt at integration. A preliminary program of this nature has, in fact, been written.[9]

If we are correct in our contention that the practice of applied mathematics is about to change radically. then we must question whether it is not possible that we are teaching the wrong skills. No sooner do we ask this question than the answer comes crashing down upon us. If the engineer will have all his future integrations performed by machine, why bother to teach him a bag of tricks that he will immediately forget? Why not teach a relevant skill? Why not teach him how to use the machine as a partner in solving his problem? While the power that the machine offers the mathematician is enormous and while a number of relevant systems are, as they say, "user oriented," it still represents a new discipline for the mathematician to master. *The first course in on-line symbiotic symbolic problem solving in applied mathematics has, in fact, already been given.* It was presented within the Applied Mathematics Group of the Mathematics Department of MIT during the 1969-1970 academic year by Professor Michael Fischer. It is a source of pride to us that the computational vehicle for this course was our MATHLAB program.

An instinctive reaction against the bag-of-tricks philosophy of teaching calculus was already present many years ago. Since integration in terms of elementary functions was a mess, would it not be better (or at least more elegant), it was asked, to teach the rigorous foundations of analysis? So a lot of deltas and epsilons were substituted for special-purpose computational tools. This is quite similar in spirit to the post-*Sputnik* fad of teaching "new (i.e., 19th century) math" in elementary school. Rather than get bogged down with special-purpose techniques, why not explore the underlying logical foundations? Unfortunately, this did not do the physicist any good. He did not need existence theorems; he needed computational facility.

Fortunately, we can now see our way out of this dilemma. For one thing, symbolic integration need no longer be regarded as a bag of tricks. This is due to remarkable strides[10] having been made toward the provision of a decision procedure for the problem of expressing indefinite integrals in terms of elementary functions. So, the stigma has been removed. Symbolic integration is now respectable mathematics. Furthermore, these remarks may also be applicable in spirit to all the familiar processes of analysis.

Emerging from all this, there would seem to be a natural division of instruction. For the physicist and the engineer, it would seem most appropriate to present courses geared to the coming technological developments. They should be taught to think in terms of solving their problems in consort with machines. For the applied mathematician, the emphasis might be on the comprehension of computational processes as algorithmic. They might then be expected to contribute to our computational repertoire by extending the scope of our present algorithms or, as is also necessary, by improving their efficiency.

The question of efficiency is an interesting one. New algorithms[11] for computing the greatest common divisor of two polynomials have been discovered which are orders of magnitude faster than the one attributed to Euclid. What is intriguing is that Euclid's method could have stood essentially unchanged for 2300 years. The reason for this is that, prior to the advent of high-speed digital computers, the main purpose of this and similar algorithms was not for actual computation. These algorithms were desired to lend a sense of reality to the existence results; constructive existence proofs leave us with none of the apprehensions attached to nonconstructive ones. Except for low-order cases, it was not possible to consider actual computation. Now that such computations are conceivable, the search for efficiency is on.

Similar comments can be made regarding improvements [12] to the classical algorithm of Kronecker[13] for the factorization of polynomials over the integers. Here, perhaps, is the most telling evidence of the inadequacy of our current curriculum. Almost all of us have gone through high school, college, and graduate school—and this refers to mathematicians, pure and applied, as well as to those in the physical sciences—factoring hundreds, if not thousands, of polynomials without once asking if an algorithm exists.

It should be obvious, by now, that the author is an enthusiast. It might then be reasonable to ask whether there are not, perhaps, lessons to be learned here which are independent of the rate at which the anticipated developments actually proceed. A number of the participants in this conference commented on the difficulty of teaching calculus to undergraduates. Yet, to a not negligible extent, we have succeeded in teaching calculus to machines.

We have no desire to minimize the differences between students and computers, the lack of mathematical preparation with which students often arrive in college, or the extent to which their elementary and secondary educations might have actually wounded them in their capacity for learning mathematics. Nonetheless, we feel we must ask whether the algorithmic approach, so successful in programming computers to perform symbolic mathematics, might not prove of advantage in teaching undergraduate mathematics. Rather than venture an opinion, we would prefer to see the experiment tried.

APPENDIX: INFORMATION FOR PROSPECTIVE USERS

Concerning the implementation and availability of MATHLAB, the system is a modular program written in LISP. Among the more important modules are those for parsing the input string, for the two-dimensional display of MATHLAB results, for general symbolic simplication and differentiation, for the seminumerical computation of rational functions, for integration, for direct and inverse Laplace transforms and for the solution of linear differential equations with constant (symbolic) coefficients. The action of these several modules must be directed by a rather large supervisory program whose duty is to interpret the user's requests and to call upon the services of the various modules at the appropriate times, presenting each with the data in a form most appropriate to its particular set of algorithms.

Putting all this together as a program implemented within a time-sharing system on a Digital Equipment Corporation PDP-10 computer, allowing for the original LISP substructure, and for a modest core requirement (about 10K) for the user data and computation space, we find we require about 53K thirty-six bit words of core. It is possible to move this number a little in either direction by altering the facilities provided. It is possible to increase it enormously in a usually vain attempt to effect an extremely large computation.

As for time estimates, the time required for each step in the Poisson brackets example is probably less than a second; the differential equation example takes something more like fifteen seconds; most of this is time involved in the internal factorization of a polynomial of degree five, in five variables, into a linear polynomial and two irreducible quadratics.

We are currently (February 1971) preparing a version of MATHLAB for release through DECUS (the Digital Equipment Computers Users Society).

For any institution possessing a PDP-10 or access to one through a commercial time-sharing system which permits users large blocks of core, this is clearly the preferred route. We have, in the past, provided programs and some aid to a few people who wanted to implement parts of MATHLAB within the LISP systems of other computers. Procedures such as the factorization of polynomials, the integration of rational functions, and our programs for the two-dimensional display of mathematical expressions have proved to be particularly popular. Success has been reported on the IBM system 360 and the CDC 6600. Any future request would have to be entertained on an individual basis.

REFERENCES

1. C. Delaunay, *Mem. Acad. Sci.*, *28*, 883 and *29*, 931 (1867).
2. A. Deprit, J. Henrad, and A. Rom, "Lunar Ephemeris: Delaunay's Theory Revisited," *Science* June 26, 1970.
3. C. Engelman, "MATHLAB 68," in A.J.H. Morrell (ed.). *Information Processing 68*, North-Holland, Amsterdam, 1969, pp. 462-467.
4. C. Engelman, "Varieties of Interactive Experience in Machine Aided Mathematics," *Digest Record, Joint Conference on Mathematical and Computer Aids to Design*, Anaheim, 1969.
5. J. E. Fromm, *Finite Difference Methods of Solution of Nonlinear Flow Processes with Application to the Benard Problem*, Los Alamos Scientific Laboratory, 1967.
6. E. T. Whittaker, *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, Cambridge University Press, Cambridge, 1927, 3rd ed., p. 311.
7. E. T. Whittaker, *op. cit.*, pp.299-300.
8. E. T. Whittaker, *op. cit.*, 1937, 4th ed., p. 311.
9. J. Moses, *A Program for Drilling Students in Freshman Calculus Integration Problems*, Memorandum MAC-M-369, Project MAC, M.I.T., March 1968.
10. R. H. Risch, "The Problem of Integration in Finite Terms," *Trans. AMS*, May 1969.
11. G. E. Collins, "Subresultants and Reduced Polynomial Remainder Sequences," *J.A.C.M.*, January 1967.
12. E. R. Berlekamp, "Factoring Polynomials over Finite Fields," *Bell System Technical Journal*, October 1967.
13. B. L. Van der Waerden, *Modern Algebra*, Frederick Ungar, New York, 1949, Sec. 25.

II

COMPUTER GRAPHICS

Introduction

One type of interactive terminal is the cathode-ray-tube (CRT) display. The most advanced CRT's use 2^{20} to 2^{27} addressable points (rasters) on a cathode-ray-tube face; the computer can be programmed to display a point, a character or a line between two points. The programmer can interact with it through the CRT console in at least three ways: (1) input alphameric information from a typewriter keyboard; (2) initiate various predefined operations on the display through a function keyboard; (3) supply graphic information directly to the CRT through electronic analog devices such as the lightpen, electronic tablet, "mouse" and "joystick." Of these, the lightpen is the most commonly used device. It contains a photocell which detects the passage of the electron beam across the CRT screen, returning very valuable information about the time and hence the location of the beam detection.

The most expensive and sophisticated CRT's contain all three modes of interaction. Their displays are continuously regenerated (refreshed) from information stored in buffers which can be filled and altered from the computer. A second, less expensive, type of CRT is the "storage" CRT in which the image, once drawn, remains until the "erase" button is pushed, either by the computer or by the viewer. Thus, the problems of refreshing the screen are not present and the amount of information flow is very much reduced, so that time-sharing of such terminals is more practical and less expensive by an order of magnitude than for the all-purpose CRT's. Although storage CRT's lack the capability for dynamic interaction through a lightpen, they can accomplish a good measure of both the graphic and alphameric capabilities of the larger systems, are readily adaptable to science education and are not prohibitively expensive, their costs being of the order of \$3000–\$10,000. Hence, with the exception of the computer-animated film by Ogborn, Hopgood and Black, the papers in this section discuss the educational uses of storage CRT's exclusively.

Two such storage-type graphic display systems have been used in educational contexts. The original one was developed by Glenn Culler and Burton Fried at the University of California; the second, THE BRAIN, was developed by Anthony G. Oettinger, at Harvard University. Both systems are based on a typewriter keyboard combined with a larger functional keyboard whose keys indicate mathematical and graphic operations which the system can perform. Both take the viewpoint that the user is working with arrays of real or complex numbers within the system, but the beginning user can work as if he were manipulating and displaying functions. Thus, with a few key-pushes he can exhibit a function, display its derivative, integrate it, etc. The operation of these two systems is described in the papers by DesMaisons *et al.* and by White and Fried.

Many other useful educational applications of CRT's are also described in the papers by Merrill and Kelley. Both use the Tektronix 4002 storage CRT as well as more conventional "languages" than in the case of the first two papers. One drawback to the use of graphics terminals, the lack of permanent "hardcopy," is eliminated by Merrill through the use of an *XY* plotter connected to the teletype. Of course, hardcopy can be obtained through the noninteractive use of CRT's as "microfilm recorders," with a camera attachment controlled by the computer and used to photograph each display as the computer constructs it. By varying parameters from picture to picture one builds a sequence of related images and thus a computer-animated film. Hopgood's paper discusses a very imaginative application of computer animation to the random-number simulation of the attainment of equilibrium in the energy distribution of an Einstein model of a crystal.

Taken as a whole, the papers in this section are noteworthy for their treatment of a variety of interesting problems. However, the reader may have already begun to sense the pervasiveness of the ubiquitous harmonic oscillator and Rutherford scattering. We shall encounter them again. One compelling reason for the publication of these *Proceedings* was partially archival: to attempt to collect these applications in one volume, so that future science teachers will not be tempted to continuously reinvent the wheel but will be able to go on to other less well-known but equally interesting problems.

THE BRAIN System

ROBERT E. DESMAISONS, MAURY P. HEPNER, and ROBERT J. DIRKMAN

THE SYSTEM

THE BRAIN is an interactive computer system oriented toward mathematical analysis with keyboard input and graphic output, including alphameric characters. A terminal of *T*he *H*arvard *E*xperimental *B*asic *R*eckoning *A*nd *I*nstruction *N*etwork consists of a storage tube scope and a combination function and typewriter keyboard. (Figure 1.) Before discussing the function-

FIGURE 1 A terminal of THE BRAIN.

Aiken Computation Laboratory, Harvard University, Cambridge, Massachusetts. This research has been supported in part by the National Science Foundation under Grant NSF-GY-6181 and by the Bell Telephone Laboratories, Inc., under a contract with Harvard University.

al appearance of the system to a user, we shall review some of the developmental background of the system.

Project TACT (*T*echnological *A*ids to *C*reative *T*hought) considered the role of modern technological aids in the teaching and learning processes. Since the computer was among the most promising technological aids available, a study was made of interactive systems that (1) provided graphical output, (2) possessed a language that could enable the user to communicate with the machine with little previous programming experience, and (3) was suitable for use in the classroom.

At the time (1965), the only system that came close to matching those criteria was one developed by Glenn Culler and Burton Fried at the University of California at Santa Barbara. Arrangements were made to install two terminals of this system at Harvard, connected via Western Union lines. These were then used in statistics and biology courses. Although users were generally satisfied with the language and "feel" of the system, they were nevertheless unhappy with its performance and flexibility for several reasons. Among these were the loss of reliability inherent in the numerous links in our communications chain, the fact that we could not change the system to conform to our particular usage, and our inability to move it, due to lack of documentation, to another machine within the Harvard environment.

While the Culler-Fried terminals were being used, a simulator of that system was implemented under the Compatible Time Sharing System (CTSS) of Project M.I.T. This served as a test-bed for systems experimentation and a design exercise for the eventual construction of THE BRAIN. However, the simulator had the same limitations as any other user of CTSS in that response time and accessibility to the computer were very poor unless one were willing to work between midnight and 8:00 a.m. Although systems people can sometimes accept those hours as a fact of life, it would not do in the classroom. Therefore, we designed our own system, first implemented for an IBM 360 model 50, and now operational under OS on the Harvard Computing Center's model 65 and McGill University's model 75.

The language of THE BRAIN is similar to that of its predecessor, the Culler-Fried system.[1] One of its strongest features from a mathematical standpoint is the ability to work with functions or arrays of numbers rather than with only single values. Each user has a "working register" which may be thought of as varying in length depending upon the number of components in the array that he is creating or manipulating. The working register acts as an accumulator; its contents are used as input to a mathematical operation and are then replaced by the result. However, as an example of the flavor of the language, we can examine the following sequence of keypushes which could be used to display the function $y = \sin x$ in the range $-2\pi < x < 2\pi$. The

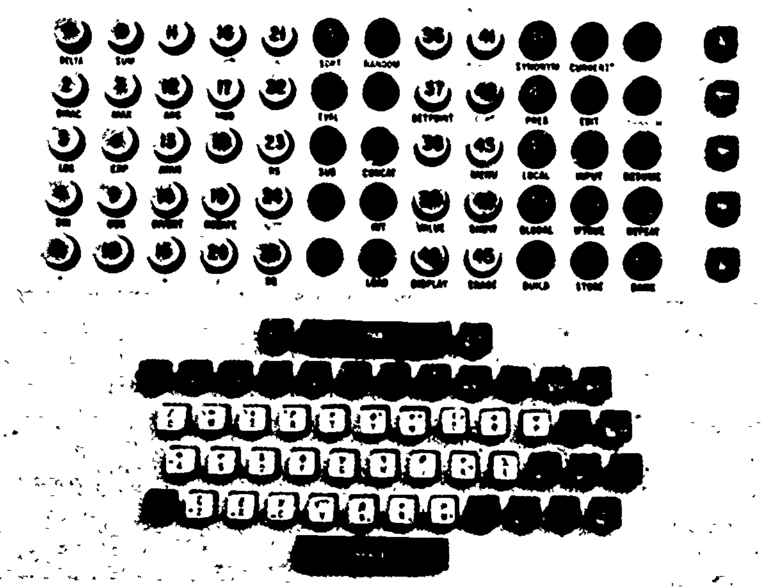


FIGURE 2 THE BRAIN keyboard.

underlined expressions can be generated by single keypushes on the upper keyboard of Figure 2. Thus the statement

INT -2π 2π 100 = X SIN = Y DISPLAY Y X

comprises the operations

(INT -2π 2π 100)–calls upon the system operator, INT, which creates in the working register a collection of 100 equally spaced values of X between -2π and 2π (note that π is on the keyboard).

(= X)–stores a permanent copy of the working register under the name X as an array of 100 values.

(SIN)–computes the sine of each component in the working register, leaving the result there.

(= Y)–stores a permanent copy of the working register under the name Y.

(DISPLAY Y X)—displays the function Y versus the function X in the Cartesian plane where Y is the ordinate and X the abscissa. The result is two complete periods of a sine-curve displayed on the cathode-ray-tube (CRT) screen.

Note that only 19 keys, including spaces, need to be pushed to obtain this result.

From our work with the Culler-Fried system it was noted that the strict use of only single character names in defining user variables and operators could become both confusing and confining. We thus decided to give the user the option of using single *buttonpush* names or multiple character names. A user of THE BRAIN can therefore be one who wants to minimize his keypushes, one who occasionally uses mnemonic names and likes to switch easily from one mode to the other, or a typist who would rather type the characters E-R-A-S-E followed by a space then lift his hand to the ERASE button.

We have also attempted to provide the user with flexible display formatting capabilities. The standard format assumes a set of default options which automatically: (1) locates the origin of coordinates at the screen center; (2) determines the scale of the display from the maximum absolute values of the X and Y arrays; (3) fills as much of the scope surface as possible while maintaining a true relationship between ordinate and abscissa; (4) draws subsequent displays (before an erase) on the same scale as the first display. However, the user can define and invoke his own format which will: (1) define any physical portion of the scope as the display area or *viewport;* (2) define the *window* or area of the mathematical function which is to be mapped onto the viewport; (3) stretch the curve in either the X or Y direction; (4) allow the origin to appear anywhere within the viewport; (5) draw subsequent displays to their own individual scales. These and other options allow the user to move the viewport, use several viewports at once, or to magnify the desired display by simply changing the format, rather than the mathematical function itself.

Another feature that uses the display scope is the *echoing* of keypushes. Each button on the keyboard has associated with it a predefined "display value." In the echoing mode, each time a key is depressed its display value is exhibited. For example, as seen in Figure 2, button #45 on the upper keyboard erases the screen and displays the characters E-R-A-S-E. Between erasures of the screen, a user in echo mode is able to look back at his recent history of keypushes; and yet if he is constructing a display and wants to eliminate any cluttering of echoed keypushes, he can easily inhibit the echo feature. If the user desires a more permanent record of his data, operators, or displays, he may store them on disk for later restoration or take away from the session a videotape, polaroid snapshots and slides, or CALCOMP output.

It is possible to store a sequence of keypushes under a given name to be later executed by calling that name as a user-defined operator. This operator can be considered a subroutine; the user may pass arguments to it, declare variables as local during its execution, and perform branching, conditional or unconditional, within the operator. User-defined operators may be nested within one another and can be non-destructively interrupted during execution so that the user may return to manual mode, examine intermediate results and then resume execution of the operator at the point of interruption.

The Culler-Fried type of language gives the user the ability to construct a function "from the inside-out," i.e., building the function part-by-part in the order of computation. For example, to calculate the function $y = \sin 20x + \cos 10x$, the consecutive keypushes might be:

$$\underline{\text{LOAD}}\ X \cdot 20\ \underline{\text{SIN}} = \text{Temp}\ \ \underline{\text{LOAD}}\ X \cdot 10\ \underline{\text{COS}} + \text{Temp} = Y$$

In many cases this inside-out method of operation is useful in understanding the function and sometimes in actually directing its construction. But the user may know beforehand exactly how the function is to be specified and might prefer to define the function (from the outside-in) via a FORTRAN-like expression. With THE BRAIN a user can substitute a parenthesized expression at any place where a number or an array is expected. The system then parses this according to a simple precedence scheme and can either leave the result in the working register or perform assignment as part of the expression. In the preceding example, the outside-in form of usage would result in:

$$(Y = \underline{\text{SIN}}\ 20\ X + \underline{\text{COS}}\ 10\ X)$$

The user is thus given considerable flexibility in constructing functions.

Perhaps the keyword for describing THE BRAIN is "flexibility." This design attitude has been applied not only in making the system comfortable from a human engineering viewpoint but also in providing for modularity and complete documentation in the system. With these it is hoped that it can be modified internally to adapt to its users' needs and earily exported to other installations.

Readers of this paper who might be interested in usage of THE BRAIN at locations other than Harvard could consider two options.

(1) Complete exportation of the software and hardware, requiring:
  - (a) An IBM 360 model 50 or higher with external interrupt feature.
  - (b) An IBM 2703 or 2702 transmission unit.
  - (c) THE BRAIN Keyboard Control Unit (controls up to 256 terminals and costs about $15,000).

(d) THE BRAIN terminal (Keyboard and Display unit)–costs about $6000 per terminal.

(e) Copy of the software–no cost.

Items (c) and (d) can be constructed at Harvard in approximately 6 months time.

(2) Exportation of a terminal of THE BRAIN, requiring:

(a) Construction of a terminal (about $6000 each).

(b) Establishment of a full duplex private telephone circuit or dial-up system between Harvard and the location of the terminal.

Both options have been tried successfully, the first with two terminals operating at McGill University in Montreal on their IBM 360/75, and the second with a working terminal at the Harvard Business School, which is across the Charles River from the Harvard Computing Center.

In terms of resources used, an average user might require 2 or 4 minutes of CPU time during one console hour. The basic software uses about 160K bytes of storage, while each logged-in terminal demands an additional 50K bytes.

TEACHING ELEMENTARY STATISTICS WITH THE BRAIN

There is a wide range of instructional settings in which the computer has been utilized with varying degrees of success. On one end of the spectrum, the absence of a teacher and any predesigned materials is characteristic of most computational uses. On the other end of the spectrum, the setting is a teacher-centered classroom. Here the instructor uses the computer in a sense as an animated blackboard to present ideas and enhance communications. The remainder of the continuum finds settings reflecting varying degrees of teacher influence and control through such designs as computerized programmed instruction, simulations, games, etc.

The following is an account of an experiment conducted by Anthony G. Oettinger of Harvard University and Maury P. Hepner which examines situations arising at the two ends of the spectrum. Our basic hypothesis was that the various forms of communications technology are providing us with potentially new and different modes of expression and communication. We set out to investigate these potentialities by using THE BRAIN as one aid in the teaching of selected topics in elementary statistics.

The statistics material was presented in a four-period sequence of an on-going course in computational linguistics held in the fall of 1969. The sequence was designed to enable students to decode cryptograms on the basis of letter frequency counts. The material included such topics as frequency distributions, populations and samples, the chi-square statistic, and tests for a null hypothesis.

The class was held in a conference room with the following: a terminal of THE BRAIN, two large television monitors for viewing THE BRAIN output, a video camera, a video recorder, a lantern slide projector, a projection screen, a blackboard, and twenty Harvard and Radcliffe undergraduates. The students were not instructed in the use of THE BRAIN, and the instructor made no attempt to explain his keyboard operations except in times of programming difficulties.

The first class period introduced the students to the statistical concepts of mean and variance. In a previously assigned homework set, the students had been asked to count the frequency of occurrence of each letter of the alphabet for given French and English texts and to record the average time taken per letter count for each text. The class period was mainly spent displaying and discussing the student data points (average time per letter) for the six different given texts. THE BRAIN output as viewed on the TV monitor recorded student results for the six texts. These results were then examined in terms of the mean and spread of the times as calculated on THE BRAIN.

The second class period focused on ways of characterizing and analyzing the properties of the letter counts with histograms of the frequency and cumulative frequency distributions. THE BRAIN was used only briefly to generate a few of the distributions. The third class introduced two fundamental distributions: the rectangular distribution and the chi-square distribution. They were developed almost solely on THE BRAIN. The rectangular distribution was motivated by comparison of the "cumulative distributions" of a random sample and an equal-sized uniform sample. The chi-square statistic was given a similar heuristic development.

The fourth and final class was devoted to the development and use of the theoretical chi-square distribution for testing the validity of the assumption of a given parent population. THE BRAIN was used to generate the chi-square frequency distribution and to determine graphically the 95% probability level. Slides, previously taken from THE BRAIN outputs, were then used to show sample chi-square values taken against correct and incorrect parent populations.

Learning to use THE BRAIN to construct the required functions for this project took one author (Hepner) roughly 15 hours. Since the resulting displays embodied a certain element of discovery it was erroneously assumed that the students would be led by the displays to experience these discoveries. Consequently, much time was spent in combining elementary operations into complex operators, enabling the generation of highly polished displays at the push of one key. The motivation was elegance and efficiency, and the final product was compact; but the level of nesting made the operator highly resistant to any changes.

The excitement generated by our individual discoveries was sustained in joint planning sessions. The result of these investigations was a wealth of graphic materials and examples for classroom demonstrations and homework assignments. However, it became apparent that the mode of operator construction or packaging would be a determining factor in determining whether the class could follow the instructors' use of THE BRAIN, let alone lead its development. Hence, in addition to highly polished, but somewhat inflexible, complex display operators we also developed their component operators: some for computation, some for display generation, some for axes construction, etc. Much of our final classroom resources consisted of operators in this form, often with parameters left variable so that the operator remains flexible. While this introduced new areas of application, the use of these packages of operators became intellectually more demanding and difficult to use under classroom pressures.

From the outset of the experimental lectures it was clear that we were not stimulating any great response from the class. The excitement and discovery of the planning sessions were totally absent. The instructor tried to use THE BRAIN as an animated blackboard, to present information, to develop concepts, to illustrate examples, etc. Prepared with a battery of operators, he spent a large portion of each class pushing keys and discussing the results. However, the operators were difficult to recall and combine spontaneously, and the time taken to generate displays was invariably too long. In one class, the instructor spent a full 34% of the time silently pushing keys. The average time to develop a single display was slightly under one minute.

The amount of overt class participation was negligible. It appears that the students had difficulty following the displays. Often material appearing on the monitors was a reflection of the key pushes occurring and not of the statistics context. The students were burdened with irrelevant and incomprehensible cues, while the instructors were often motivated not by class response but rather by the computer response. While generating displays, the intermediate results induced us to go off in new directions and explore new possibilities. In private, outside the classroom, this is the meat of discovery. In the classroom, this is trouble.

Our two basic uses of THE BRAIN—as a tool for individual problem solving and as an animated blackboard—met with very different degrees of success. Our individual efforts were personally rewarding, but our classroom efforts were dismal. To attempt to improve upon the unsatisfactory use of THE BRAIN in the classroom, attention should now be brought to bear on the questions of packaging and graphical communication.

INDIVIDUAL INSTRUCTION WITH THE BRAIN

This section is concerned with a program being used for research in individual instruction techniques using THE BRAIN. The program is a graphical computational tool for undergraduate engineering education, designed primarily for interactive use by the student. There are two primary research objectives for the development and use of this program: (1) to identify the principal factors involved in the design of a computational tool suitable for interactive instruction and (2) to provide a vehicle for studying instructional strategies with various amounts of opportunity for exploration by the student.

Linear dynamic systems can be characterized by specifying two sets of complex numbers, the poles and zeros of the system. A configuration of poles and zeros on the complex plane (the s-plane) corresponds within a multiplicative constant to a unique function of time. Because this s-plane representation is the basis of powerful techniques for analysis and synthesis of linear systems, it would be very desirable for the students to be facile in dealing with these correspondences. Usually, the relations between s-plane configurations and time functions are not explored in much depth, because the effort required in obtaining corresponding pairs by hand is so time-consuming that the student obtains little experience in dealing with them.

The instructional strategies in the study are addressed to this problem. We seek the most useful instructional approaches, using THE BRAIN system, to aid the student in learning the relationships between pole-zero configurations and corresponding time-dependences. If the computer allows exploration of system properties, rather than just exposure to the analytical techniques required to treat them, then this adds a new dimension indeed to the study of these systems.

The pole-zero to time-domain transformation program generates the inverse Laplace transform of a class of functions in s. This class is comprised of rational functions with real coefficients $X(s)$.

The roots of the numerator are the zeros (z_i), and the roots of the denominator are the poles (p_j). Hence we can express $X(s)$ as

$$X(s) = K \prod_{i=1}^{n_z} (s-z_i) \Big/ \prod_{j=1}^{n_p} (s-p_j)$$

where n_z is the degree of the numerator (the number of zeros), n_p is the degree of the denominator (the number of poles), and K is a real constant.

We put two constraints on $X(s)$ for the purposes of the computational program:

1. $X(s)$ can have only simple poles ($p_j \neq p_k$ for $j \neq k$);
2. The denominator of $X(s)$ has a degree at least one greater than the numerator, i.e., $X(s)$ has at least one more pole than zero.

The first constraint greatlv simplifies the procedure for calculating the time function. Multiple poles can be approximated satisfactorily by making the several poles very nearly equal. The second constraint insures that there are no impulses or doublets in the time function; these would likewise overly complicate the program.

Under these constraints the partial fraction expansion of $X(s)$ gives

$$X(s) = K \sum_{j=1}^{n_p} \frac{c_j}{(s-p_j)}$$

where, for $s = p_j$,

$$c_j = \prod_{i=1}^{n_z} (s-z_i) \Big/ \prod_{\substack{k=1 \\ k \neq j}}^{n_p} (s-p_k)$$

then the time function is given by

$$x(t) = K \sum_{j=1}^{n_p} c_j e^{p_j t}$$

If p_j is real, there will be a real exponential term $c_j e^{p_j t}$; if p_j is complex, there will be a term corresponding to $c_j e^{p_j t}$ of the form $\bar{c}_j e^{\bar{p}_j t}$ where the bar denotes the compex conjugate, and the sum of these two terms will give a function of the from $Ae^{bt} \sin(\omega t + \phi)$. As the time function $x(t)$ is being calculated, components corresponding to real poles and complex conjugate poles are displayed as dotted curves as they are generated.

The design of an interactive program involves a tradeoff between flexibility and simplicity. On one hand it is desirable to allow many directions for exploration. Ideally, the program should be capable of providing answers to any question the user may ask. However, such an objective would make enormous demands on the program. For realistic input-output design and response times, it is necessary to constrain the universe that the user may consider.

The design of the program requires consideration of questions such as the following:

What queries is the student allowed to pose?

The program allows the user to generate the time function corresponding to any pole-zero configuration on the s-plane provided the number of poles is greater than the number of zeros and the poles are all simple. The time functions are stored so that they may be recalled and compared with other time functions.

How difficult is it to pose and modify the query?

The input procedure employs the capability of the system to invoke an operator by pushing a single upper keyboard button. All the operators that the user may access are associated with upper keyboard buttons as shown in Figure 3. Some of the operators require one or more arguments such as numerical data that are input on the lower (typewriter) keyboard.

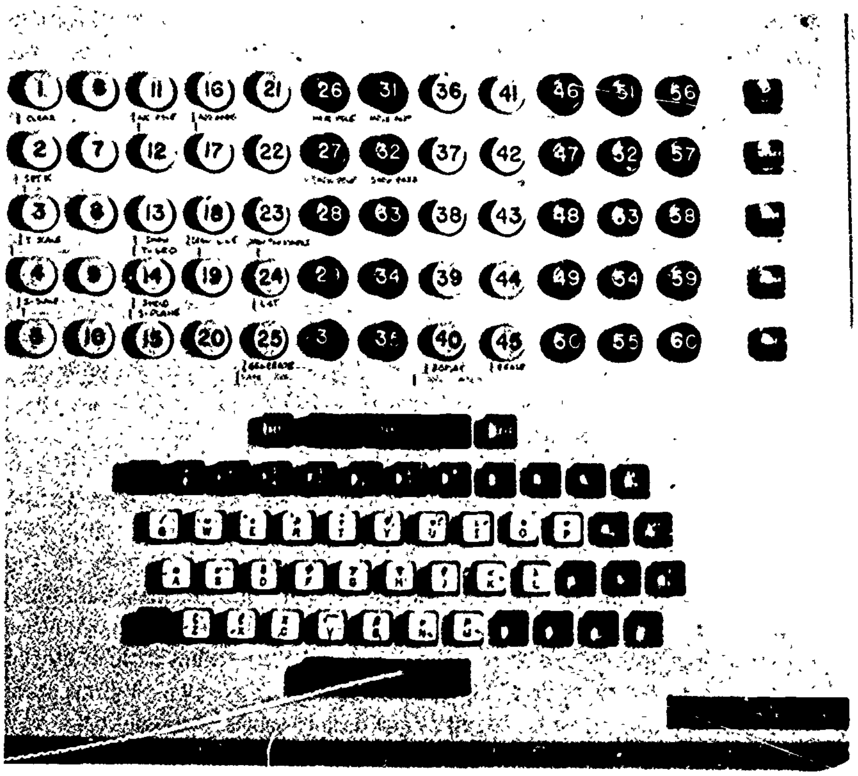


FIGURE 3 User operators for Laplace transform problem.

The operators fall into four classes:

(1) initializing operators which clear the pole and zero registers, set the multiplicative constant K, define the time domain, and set the scales for the time and s-plane displays;
(2) display operators which are used to compose displays;
(3) pole and zero manipulation operators for adding and moving poles and zeros;
(4) time function generation operator which generates the time function corresponding to the current poles and zeros and value of K.

How long must the user wait to get the response?

The time required to generate the time function was excessively long for interactive use (up to 60 seconds). To make this waiting period less objectionable, it was found very instructive to display the real functions corresponding to each pole and conjugate pole pair in the partial fraction expansion as they were generated. This long response time is still undesirable, and the program has been recoded to reduce the generation time by a factor of three or four. The other operators have adequately short response times.

How clear is the response?

The design of the output format is very critical from an instructional point of view. The displays should be flexible and easy to compose to simplify exploration and very clear to minimize the possibility of misinterpretation. The display has three parts: (1) a time plot, (2) an s-plane plot, and (3) a listing of the poles, zeros and constant, K.

Since the magnitude of a time function is generally not known until it is displayed, it is necessary to be able to change the window (the range of the ordinate and abscissa) of the time axes very easily. This is accomplished by the operator T-SCALE, which has three arguments corresponding to the minimum and the maximum values of the ordinate, and the maximum time. The window may be changed at any time by using T-SCALE. Another operator displays the time axes and exhibits the three values defined by the current scale. Any number of tickmarks may be displayed on the time axes, and horizontal lines may be added at any desired values of the ordinate. Any previously generated time function may be plotted on the current time axes by using the DISPLAY operator.

The s-plane plot is simpler. The origin is always in the center of the plane, and the window is sized by a single argument to the S-SCALE operator. The

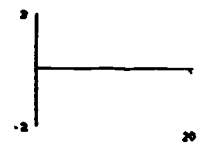
CLEAR (clears pole and zero arrays)

SET K 1 (sets the multiplicative constant, K, to 1)

T-SCALE -2 2 20

(sets up the scale for the time domain display and creates a time array of 101 points for $0 \leq t \leq 20$ which will be the domain of any time functions generated before T-SCALE is invoked again)

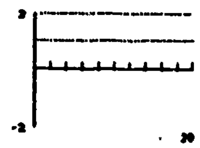
SHOW T-GRID



DRAW LINE 1

DRAW LINE 2

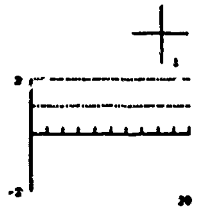
DRAW TICKMARKS 10



S-SCALE 1

(sets up the scale for the s-plane axes)

SHOW S-PLANE



ADD POLE -.15,1

(adds the complex pole pair -.15+j1 to the pole array)

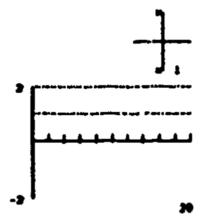
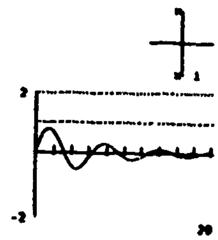


FIGURE 4 Example of the Laplace transform program.

GENERATE a A

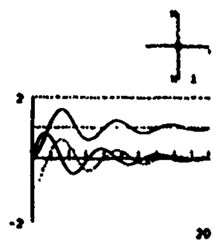
(generates the time function corresponding to the current poles, zeros, and value of K and assigns it the name 'a'; the current time domain as defined by the last T-SCALE is assigned the name 'A')



ADD POLE 0

GENERATE b A

(each real component of the partial fraction expansion is also plotted when it is generated; the function 'b' is hence the sum of the damped sinusoid shown dotted, corresponding to the pair of complex poles, and the unit step function, corresponding to the pole at the origin)



ERASE

SHOW T-GRID

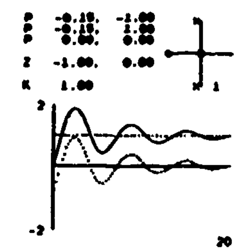
SHOW S-PLANE

(current poles and zeros are also plotted)

ADD ZERO -1

LIST

GENERATE c A



ERASE

T-SCALE -1 2 10

(a new scale for the time axes is set up)

SHOW T-GRID

DRAW LINE 1

DRAW TICKMARKS 10

DISPLAY () a A

DISPLAY () b A

DISPLAY c A

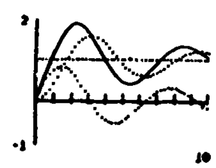


FIGURE 4 (Continued)

operator SHOW S-PLANE draws the axes and displays the current poles and zeros. Adding or moving poles and zeros is automatically indicated on the *s*-plane. A listing of the current poles and zeros and the value of K can be made at any time using the operator LIST.

The interactive use of the program is illustrated in Figure 4. A damped sinusoid is generated (corresponding to the complex pole pair at $-.15 \pm j$), then its integral is constructed (by adding a pole at the origin), and finally the integrated function is modified (by adding a zero at -1). These three functions are displayed together on an expanded time plane in the last frame. Notice that the third function is the sum of the first two.

REFERENCE

1. R. E. DesMaisons and K. B. Winiecki. *THE BRAIN Users' Reference Manual, Draft 15*, Project TACT Document, March 1970.

Mathematical On-Line Systems as a Tool for Teaching Physics

ROSCOE B. WHITE and BURTON D. FRIED

The mathematical on-line systems, developed by Professor Glenn Culler at Thompson-Ramo-Wooldridge and the University of California at Santa Barbara, and currently implemented on large IBM 360 installations at the Santa Barbara and Los Angeles campuses of the University of California, provide a uniquely simple means of experimentation with the use of computers in physics teaching. A description of this type of system is given in Appendix A and has also been given elsewhere.[1-4] Although developed primarily for problem solving in research applications, such systems have proved valuable for instructional purposes because of the following features: (1) highly interactive graphics via a storage oscilloscope, including graphic input; (2) negligible software overhead for *mathematically trained* users; (3) orientation of the system toward advanced mathematics (i.e., elementary classical analysis).

LECTURE-DEMONSTRATION

Using closed-circuit TV or other conventional methods of projecting the oscilloscope traces for classroom viewing, the instructor illustrates some features of a problem or problem area. This can be superior to a film if and only if there is real interaction between instructor and students; i.e., if the classroom ambiance is such as to encourage questions of the "What if..." variety. These may (and it is hoped will) involve some points not anticipated by the instructor, but the on-line construction of console programs is so simple that it is completely feasible for the instructor to explore, with student participation, new aspects or variants of the problem—at the same time giving the students an invaluable example of how problems are really attacked.

An eigenvalue problem, such as Schrödinger's equation, provides a good

Department of Physics, University of California, Los Angeles, California.

example of this mode. In suitably dimensionless form we have (in one dimension)

$$d^2\psi/dx^2 + [E - V(x)]\,\psi = 0 \tag{1}$$

where the potential is assumed symmetric, $V(x) = V(-x)$, so that we can work with even or odd functions. (Parity is a good quantum number.) To start with, consider the case of a square well.

$$V(x) = \begin{cases} -V_0, & 0 < x < a \\ 0, & a < x \end{cases} \tag{2}$$

in which case an analytic equation for the eigenvalue can be given:

$$\sqrt{-E} = \begin{cases} k_0 \tan k_0 a, & \text{even parity} \\ -k_0 \cot k_0 a, & \text{odd parity} \end{cases} \tag{3}$$

$$k_0 = (V_0 + E)^{1/2} \tag{4}$$

equivalent to the implicit equations

$$\begin{aligned} f_e(E) &\equiv k_0 \sin k_0 a - \sqrt{-E}\cos k_0 a = 0, \text{ even parity} \\ f_0(E) &\equiv k_0 \cos k_0 a + \sqrt{-E}\sin k_0 a = 0, \text{ odd parity} \end{aligned} \tag{5}$$

We find the lowest roots of (5) by displaying $f_e(E)$ and $f_0(E)$ over some range of E. This is illustrated in Figure 1 for $V_0 = 60$. Visual inspection shows how many zeros of f_e and f_0 are contained in that range; the values of the roots are, to good approximation (depending on the size of the range and the number of points) given by the non-zero values of $E\delta\,[f(E)]$, where δ is the Dirac-Kronecker function.* In this way, we quickly find the first few eigen-

*The δ operator is a typical example of facilities available in this type of on-line system which closely parallel operations familiar from, and important to, classical mathematical analysis but which are not usually found in computer systems. Acting on a vector of N components, y_n, $n = 1, 2, \ldots N$ (N typically of order 100), the δ operator generates a new vector which is equal to 1 if $y_n = 0$ or if $y_n/y_{n+1} < 0$ (corresponding to a zero crossing) and is zero otherwise, Accordingly, given two vectors, E_n and $f(E_n)$, we take $\delta\,[f]$ (by pushing one key); multiply the resulting vector by E; do a sorting operation (which arranges the results in ascending order); and display the first few and last few values. Since $\delta\,[f]$ will be non-zero only at zeros or zero crossing of f, any non-zero values of $E\delta\,[f]$ will be a reasonable
not originally chosen over too large a range). If we display the ordered values of $E\delta\,[f]$ in succession, stopping when we begin to get zeros, we will have a first approximation to any roots lying in the original range of E. Each of these roots can be refined, of course, by choosing a new E vector consisting of N values closely surrounding the first approximation of one of the roots and again looking at $E\delta\,[f(E)]$. As might be expected, this process converges extremely fast.

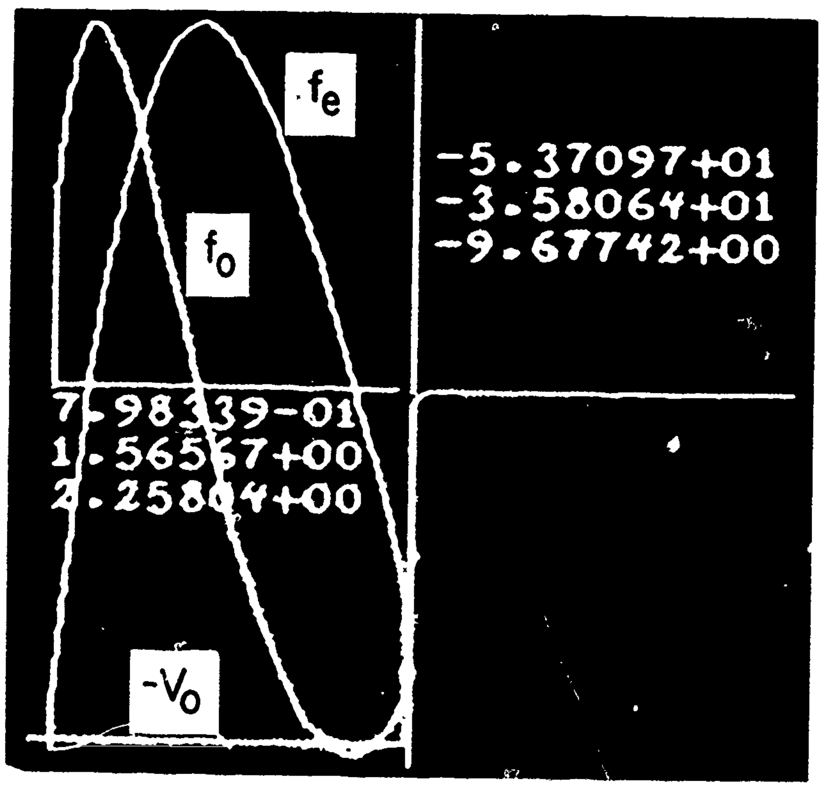


FIGURE 1 Eigenvalues of a simple square well of width $a = 0.5$, depth $V_0 = 60$ shown for $0 < x < 1$. The curves represent the implicit eigenvalue equations $f_e = 0 = f_0$; numbers in the upper right quadrant are the energy eigenvalues, those in the lower left are the corresponding values of $k_0 a/\pi$.

values of the square well for both even and odd parity and can see by inspection the validity of the approximation, $k_0 a = n\pi$, which should be satisfied for low-lying levels when $V_0 a^2 >> 1$. The lowest three roots, found in this way, and the corresponding values of $k_0 a/\pi$ are given in Figure 1.

Having found k_0 and E, it is then a simple matter to display the lowest three eigenfunctions using.

$$\psi_e = \begin{cases} \cos k_0 x \ , & 0 < x < a \\ \cos k_0 a \cdot e^{-\sqrt{-E}(x-a)} & , a < x \end{cases}$$

$$\psi_o = \begin{cases} \sin k_0 x \ , & 0 < x < a \\ \sin k_0 a \cdot e^{-\sqrt{-E}(x-a)} & , a < x \end{cases} \tag{6}$$

These are shown in Figure 2 along with a graphical representation of the eigenvalues. A different choice, $V_0 = 500$ gives the results shown in Figures 3 and 4. The increased number of bound states is apparent in Figure 3 (only the lowest three eigenvalues are listed), and, as expected, the $k_0 a/\pi$ values for the lowest three states are much closer to integers. The increased localization of the three lowest bound states is apparent from Figure 4, as is the fact that the corresponding eigenvalues now lie deep in the well.

A more interesting case for which a solution in closed form is not possible, is the rounded well,

$$V = -V_0/(1 + e^{(x-a)/b}), \quad 0 < x \qquad (7)$$

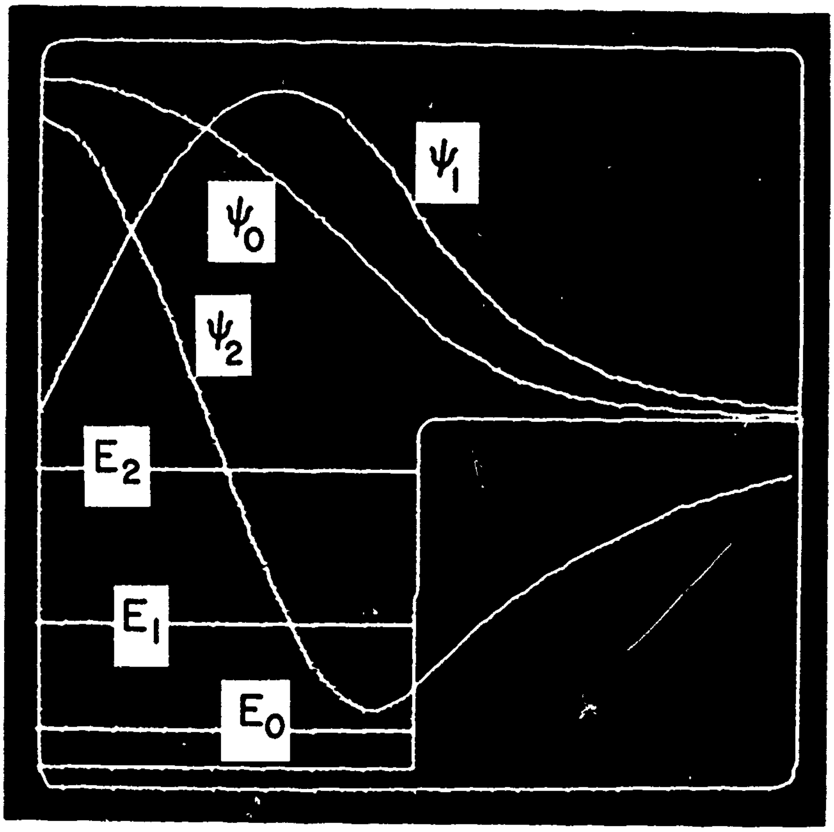


FIGURE 2 Eigenfunctions ψ_0, ψ_1, ψ_2 for the square well of Figure 1 as computed from Equation (6).

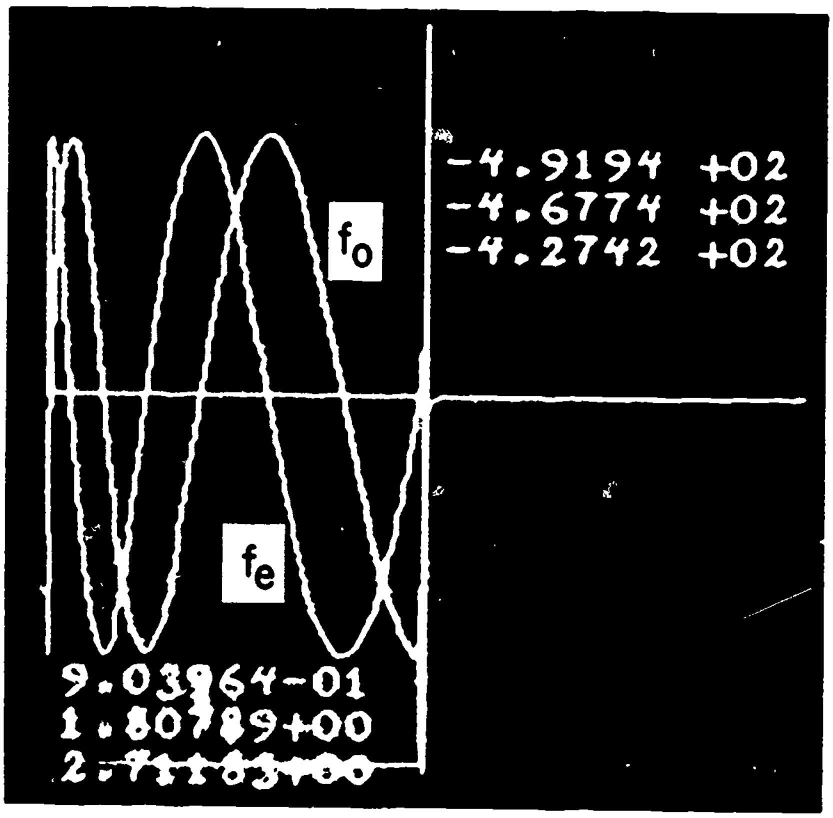


FIGURE 3 Same situation as Figure 1, but with $V_0 = 500$.

where $b << a$ gives a finite thickness to the boundary of the potential well. With any choice for E, it is easy to solve (1) by converting it to an integral equation

$$\psi(x) = \int_0^x dx' (x-x') [E - V(x')] + rx + s \equiv L[\psi(x)] \qquad (8)$$

where $(r, s) = (0, 1)$ for even parity, $(1, 0)$ for odd parity. Over some range $0 < x < c$, with $c > a$, we can solve (8) by iteration (Picard's method), i.e., by guessing a $\psi = \psi^{(1)}$ and computing

$$\psi^{(n+1)} = L[\psi^{(n)}], \quad n = 1, 2, \ldots \qquad (9)$$

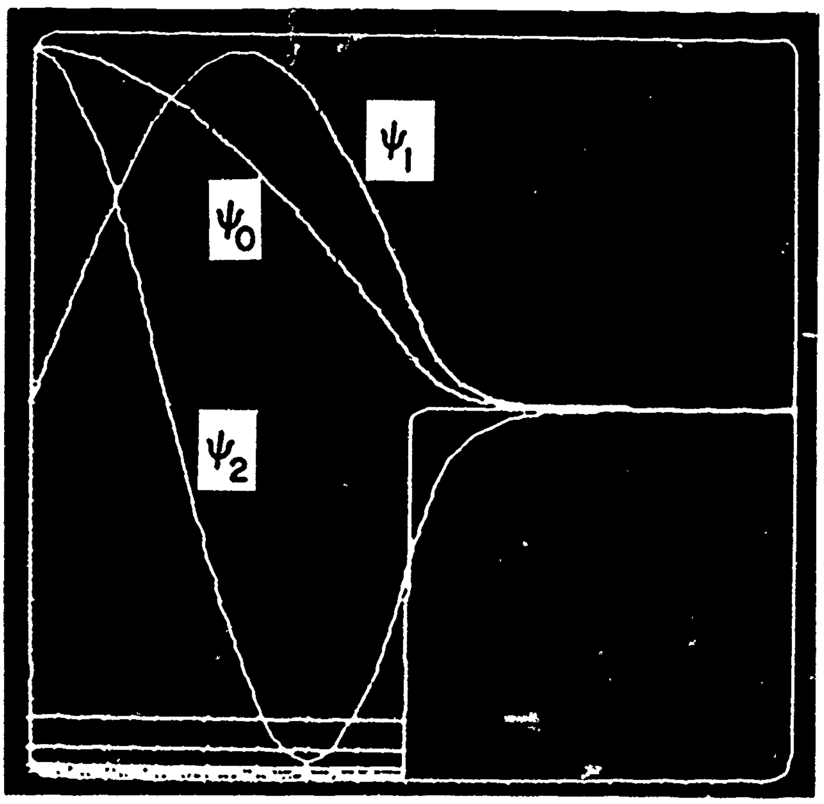


FIGURE 4 Eigenfunctions for the square well of Figure 3.

The choice of the first guess, $\psi^{(1)}$, is not critical; e.g., choosing it to be $rx + s$ is quite adequate.

The game now is to choose E so that ψ will be well behaved at large x. In practice, if one chooses c to be 2 or 3 times a it is immediately obvious from an inspection of the graph of $\psi(x)$ for two adjacent values of E how one should vary E to approach an eigenvalue (and also when one has gone too far). Moreover, unless E is close to an eigenvalue it is not necessary to iterate (3) very accurately in order to get the information needed to make the next guess at E. It is surprising how quickly this very crude procedure (which requires only two simple console programs, one for integration and one to

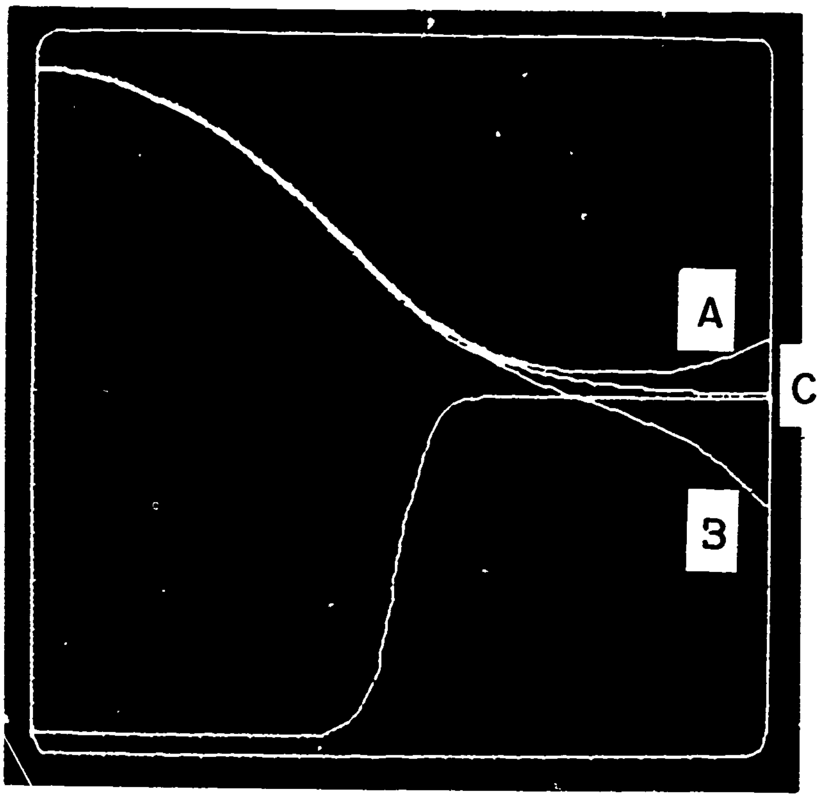


FIGURE 5 Eigenfunctions of the rounded well, defined in Equation (7), for different choices of (A) $E = -53.7097$ (lowest eigenvalue, square well), (B) $E = -53.5$, and (C) $E = -53.64$ determined by Picard's method. Curve C is the best approximation to the true eigenvalue.

compute $\psi^{(n+1)}$ from (3), given $\psi^{(n)}$) yields an eigenvalue, correct to three significant figures.

Figure 5 illustrates a rounded well, corresponding to the particular choice $a = 0.5$, $b = 0.02$ in Eq. (7), and the first few steps in the iterative determination of the lowest eigenvalue and associated eigenfunction. If we use as a first guess the lowest eigenvalue for the square well, $E = -53.7097$ the solution of the differential equation (1) gives curve A; it is clear that it will become large and positive for x larger than those shown here. [The displays encountered in iterating (8) for fixed E are illustrated in Figure 6.] As a next guess, we try a

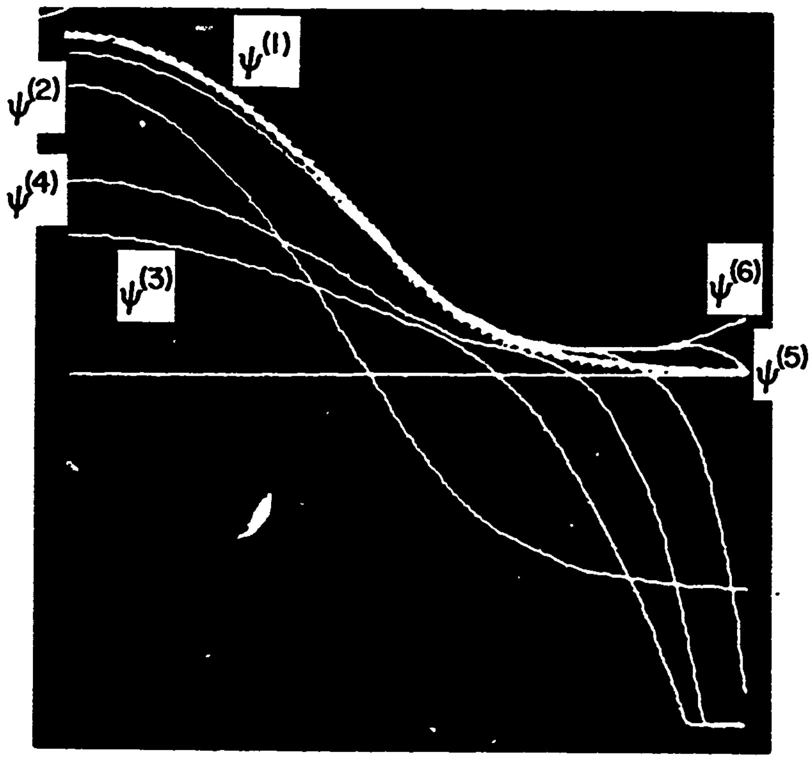


FIGURE 6 Successive iterations for eigenfunction A of Figure 5. Subsequent iterations after $\psi^{(6)}$ show no visible differences.

slightly different value, $E=-53.5$. Since the resultant solution of the differential equation (1), namely curve B, will clearly become large and negative for large x, we know that the true eigenvalue lies between -53.5 and -53.7097. A few trials show that for $E=-53.64$ we obtain the well behaved curve C, which gives us a good approximation to the true eigenvalue.

To assess the accuracy of the eigenvalue $E=-53.64$ found in Figure 5, we try the values $E = -53.65$ and $E = -53.63$. If displayed on the same scale as in Figure 5, i.e., the largest scale that will keep the whole curve within the display, we can scarcely distinguish between the curves, as we see from the

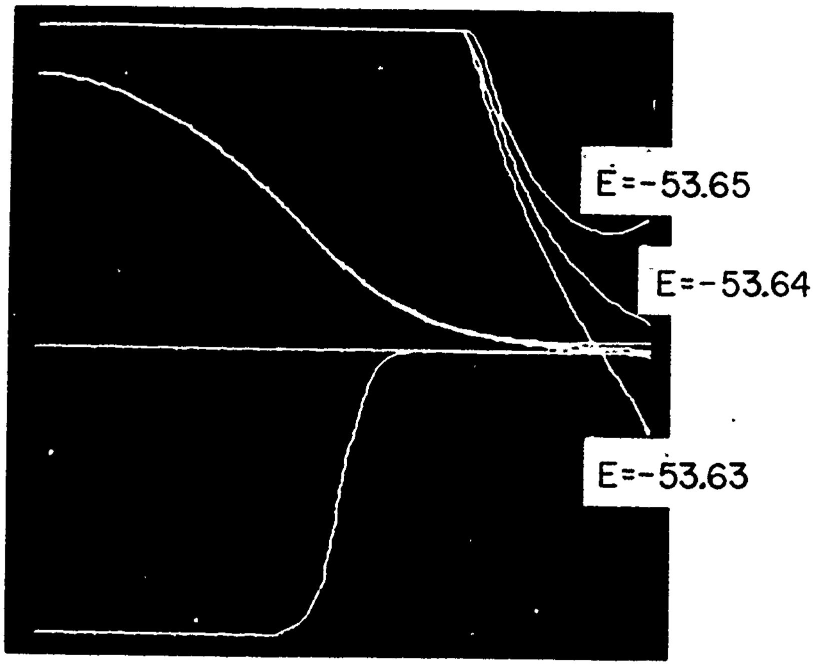


FIGURE 7 Comparison of rounded-well eigenfunctions for $E = -53.63$, -53.64, and -53.65 when drawn to the scale of Figures 5 and 6 (unlabeled) and when their tails are magnified by a factor of 2^4.

upper unlabeled curve of Figure 7 (the lower unlabeled curve is the rounded well). However, enlarging the screen scale by a factor of 2^4 gives the three eigenfunctions whose tails (labeled) are shown in Figure 7. From this it is obvious that $E = -53.65$ (corresponding to the upper curve) and $E = -53.63$ (corresponding to the lower curve) bound the correct value. Hence, $E = -53.64$ (corresponding to the middle curve) is correct within an error of ±0.01, aside from any errors associated with the number of points used in each function (here 125). The effect of this is easily ascertained by repeating the calculation for a larger number of points; since we already have a good approximation to the eigenvalue, this requires little computation.

This simple example can easily be carried much further: one can use it to illustrate perturbation theory [taking the difference between (2) and a square well as a perturbation]; variational procedures (parameterizing ψ with a few parameters and varying these to minimize the Rayleigh-Ritz integral); virtual levels (adding a centrifugal barrier term to the equation and interpreting it as a radial wave equation); etc. In any case, it is essential to keep the illustrations well within the students' comprehension, *so as to encourage the interaction between teacher and students which is the principal justification for this mode.*

STUDENT USE

In this mode of usage, the instructor prepares, in advance, a small number of console programs (typically 5 to 10) with which a student can investigate a particular problem. The student is given a description of what each program does, told what key to push, initiates each subroutine, and is asked to answer certain questions or solve certain problems. Of course, these should be structured in such a way that the more he understands about the problem – what is important, what is large and what is small, what approximations are reasonable, etc.–the easier it is for him to use the tools provided. The student need not know anything about the on-line system itself or how the console programs provided for him are actually constructed from the primitive operations of the system.

One may liken this mode to the more sophisticated of modern laboratory setups, where the student may use electronic equipment about which he knows nothing save its function and how to operate it. In general, this mode involves use of the on-line consoles in much the same way as student laboratory equipment, with a two-man "lab group" assigned for a one- or two-hour period at the console. (A typical cost, including CPU time, is $10 per console hour.) *Of particular importance* in this mode is the ease with which the instructor can prepare and modify the console program; this is a consequence of the three features mentioned in the first paragraph of this paper.

An experiment was carried out at UCLA during the spring of 1970 to teach these problem-solving techniques in a junior course in Mathematical Methods of Physics. The class of 28 students was asked if they would like spend time on the project outside of class, and the response was unanimously favorable. Rather than arrange a large room with closed-circuit television to permit viewing of the computer output by the entire class, it was decided to split the class into three roughly equal groups. Each group was given a one-hour lecture-demonstration at a computer console. It was found that this was quite adequate to permit the students to perform reasonably sophisticated problems; moreover, until a student has some minimal experience actually pushing keys, a too complicated exposition will not be absorbed.

The one-hour demonstration consisted of an introduction, with emphasis on the simplicity of the language used in the on-line systems for someone with mathematical training; a brief description of the keyboard and the various functions of its different parts; the use of the computer as a numerical calculator; operations on functions (real and complex), including generation, storage, and manipulation; the process of writing and storing a program (editing was not mentioned); and, finally, an explicit demonstration of techniques for solving a differential equation and for complex root finding, the two problems chosen in this particular case. A written mini-manual (see Appendix A) was given to the students so they could look up needed keypushes without having to search through a large manual.

The problems selected (see Appendix B) consisted of one in real analysis (solving a second-order differential equation) and one in complex analysis (root-finding). Problems were chosen to add to the contents of the course and to demand techniques of analysis taught in it. For example, the indicial equation for the differential equation must be solved in order to understand what are permissible boundary conditions. Thus there was an emphasis on man-machine interplay involving material recently taught in the course. A planned program of the type whereby the computer asks questions and the student responds was purposely avoided. This seems to us to be a good simulation of a final examination but to have little to do with the active learning process.

The students worked in pairs, each student performing the actual key pushing on one problem. A pair of students typically took 1½ hours of contact time to solve the assigned problems. They had at their disposal the programs of UCLA-1, a small collection of generally useful, simple console programs, including a second-order integration; the Cauchy complex root-finding routine, etc., although only a few of these programs were actually described to them. It was, of course, necessary to have a "laboratory assistant," familiar with the system, to take care of minor difficulties which might otherwise have wasted a great deal of time without being instructive. However, many of the problems encountered illustrated mathematical principles and proved instructive to the students. For example, in solving a homogeneous differential equation with homogeneous boundary conditions the computer will quickly converge to the functional form of the solution, but the normalization will continue to "creep" by $\sim 5\%$ each iteration, forcibly drawing the student's attention to the distinction between homogeneous and inhomogeneous boundary conditions. Several aspects of complex mapping including conformality are illustrated with great clarity, and practice is given in order-of-magnitude calculations of various types.

The student may also be taught enough about the system to permit him to construct his own programs. Naturally, this involves a certain "overhead" in terms of the effort required to familiarize the student with those system capabilities he will need. We have little experience with this mode, which is in many ways analogous to a student laboratory in which the student constructs at least some of the special equipment he needs from more elementary facilities. If that analogy is correct, then it seems likely that this mode will be better suited for instruction at the graduate level.

SUMMARY

The basic pedagogical virtue of an interactive graphic approach to teaching physics, is that it allows the student to discover at a much earlier stage just what *real* problem solving involves. Many problems customarily assigned to students are intended to drive home general analytical principles or physical laws set forth in text or lecture. However, these problems are too often artificial in that the correct choice of variable or coordinate system makes possible a solution in closed form. Only after completing his education (certainly at the undergraduate level and sometimes also at the graduate level) does he discover that a large fraction of the real world's problems lack this tidy character, and that approximations, intuition and trial-and-error are indispensable. A basic rationale for including computers of any sort must be that it allows the student some exposure to this aspect of reality. The advantage of the mathematical on-line system used here is that it provides this exposure without enmeshing the student in all the complexities that even simple computer languages entail if the environment is to have the interactive, graphic character which is so desirable for instruction in physics.

APPENDIX A: STUDENT MANUAL FOR USE OF THE ON-LINE COMPUTER

The on-line system does not require any knowledge of computer language or technique, but careful attention must be paid to the logic of the mathematical operations performed.

The Keyboard

The keyboard consists of two basic parts, the upper half consisting of keys that initiate operations ("key-pushes"), and the lower half of keys that designate storage locations. The small size of the keyboard is deceptive as it is actually several keyboards, the blue keys in the upper left-hand corner indicating different levels.

For our purposes LI is for the manipulation and storage of numbers (real and complex), and LII for the manipulation and storage of functions (real and complex). In addition to this there are various user levels. User LI and User LII contain useful programs such as integration, differentiation, contour integration, etc., which are stored under user level operator keys such as User LI +, etc. User LIII you will use for the storage of programs you write to solve your problems.

Use as a Numerical Calculator (LI Real or LI Complex)

Calculations are performed by making successive logical operations on a number placed in temporary storage. To place a number in the temporary storage location press Load, and then the number or the key of a storage position that contains the number. Now you may add to this number by using +, multiply with *, divide with /, take its sin, cos, exp, with appropriate keys of the upper keyboard. The result of your last operation will be held in temporary storage. To save a result, press Store and then the key you want the number stored under. (Key-pushes will be underlined in the text for greater clarity.) Complex numbers are typed by writing the real part, a comma, and the imaginary part. At any point of your calculation, pressing Display Return will show the contents of temporary storage.

Functions (LII Real or LII Complex)

Operations are the same as in performing numerical calculations, except that storage positions will store functions. The range of the independent variable can be determined in the following way. First place in temporary storage the Identify function by pressing LII Real ID (and Display Return if you would also like to see it). The Identify function runs from −1 to +1, on an interval −1 to +1; (0, 0) is the center of the scope. You may modify this function by adding a constant, multiplying by a number, etc. Note that the range of the scope will automatically adjust to be large enough to show the entire function. The range of the display can be found at any time by pressing Display 0 Return, and a number n will be shown meaning that the range is from -2^n to 2^n. Once you have generated the range of variable you desire, store it in the desired location. It can be displayed at any time by pressing Display and the key.

Writing a Program

To write a program, press the List key, followed by LII Real or LII Complex depending on the mode desired. Then type the operations you wish performed exactly as if you were doing the calculation. When you have finished, press List. The program can then be stored by the sequence Store User LIII and any operator key (such as sin, log) under which you wish the program stored. To initiate the program, press User LIII and the key. To see the program, press User LIII Display and the key.

Solving a Differential Equation

Decide upon the range of independent variable you are interested in, generate this range and store it under LII Real, key T. (This is convenient because the integration programs are written to use this variable.) Write a program that will solve the equation, store the result, and display it, along with the range of the scope. Enter a first approximation to the solution under its key, and initate the program. Repeat this until the procedure converges. Some time can be saved by using a reasonable approximation, i.e., one that has the correct initial behavior!

Example:

Given the equation $t d^2y/dt^2 - dy/dt + 6y = 0$, suppose we are interested in the range $t \epsilon (0, 2)$. Generate the identity, add one, and store the result under T. Introduct a second function $x = dy/dt$. Write $d^2y/dt^2 = -6y/t + x/t$. Now we see that a simple integration of this will give dy/ '. The integration routine is called by pressing User LI Sum. The program so . .s thus

List LII Real Load $X - (6 \cdot Y)/T$ User II Sum

This will give

$$\int_0^t \frac{d^2y}{dt^2}\, dt = \int_0^t dt(x-6y)/t$$

The integration routine performs the indefinite integral on the range of t stored there. Now we must add the value of dy/dt at $t = 0$, and this will give the function dy/dt. We must store this under X, and then we can integrate again to find y. Letting $dy/dt(0) = B$, $y(0) = C$, the whole program would read:

List LII Real Load $X - (6 \cdot Y)/T$
User LI Sum + LI B
Store X User LI Sum + LI C
Store Y Display Y Display 0
Return List.

Before initiating the program the desired values of B, C must be stored under their keys on LI real. In this case, the indicial equation for $y = \sum_n a_n t^{n+k}$ gives $k = 2$, so we know $y(0) = 0$ and $y'(0) = 0$. Before initiating the program, we must store initial guesses for $y(t)$ and $x(t)$. It will save a great deal of time if we use initial guesses with the correct behavor at $t = 0$, so we can use for example $y(t) = t^2$ and $x(t) = 2t$. For estimating numerical values, USER LI SQ N ENTER will surround the display scope with ticks dividing it into n equal parts.

Complex Root-Finding

For complex root-finding there are two useful subprograms. User II + will evaluate the integral

$$N = (1/2\pi i) \oint_C dT(Z)/T$$

over the contour C stored in Z for the function $T(Z)$ stored in T. User II– will evaluate the integral

$$R = 1/(2\pi i) \oint_C Z\, dT(Z)/T$$

N is the number of roots minus the number of poles of T inside the contour and R is the sum of the roots minus the sum of the poles inside the contour. To look for a root, generate a circular contour with the sequence LII Complex ID Display Return or a square contour, using LII Complex ID · Display Return. These contours may be enlarged or made smaller by multiplication and division, and moved about by adding complex numbers. Store the contour in Z. Then write a program to generate $T(Z)$ from Z, display T, display the scope size, and store T. Once a given contour is in Z and $T(Z)$ is stored in T for that contour, the two subprograms described above can be used to check whether any roots are inside the contour, and, once you have a contour containing one root, to locate it. Of course, visual inspection of the curve $T(Z)$ will show at once the value of N.

APPENDIX B: ON-LINE COMPUTER PROBLEM SET

Problem 1: Write a program to solve Bessel's differential equation and find the solution $J_n(t)$ for some n. Choose an integer $n < 10$, and to guarantee that the range of the variable includes at least the first zero of J_n take the range to be $t \in (0, 10 + n)$. When you have found the solution, photograph it and determine the value of t for which $J_n(t) = 0$.

$$t^2 d^2J_n/dt^2 + t dJ_n/dt + (t^2 - n^2) J_n(t) = 0$$

Problem 2: Find all the zeros of the function

$$T(z) = z^3 - z^2 \cdot \cos z + 3z + 1$$

for $|z| < 3$. During the search use contours consisting of both squares and circles. Write a program to calculate $T(z)$ and display it. Prove that

(a) $N = \frac{1}{2\pi i} \oint_C \frac{dT(z)}{T}$ is the number of roots of T in the contour C minus the number of poles.

(b) $R = \frac{1}{2\pi i} \oint_C z \frac{dT(z)}{T}$ is the sum of the roots enclosed in the contour C minus the sum of the poles.

Sample Solution, Problem 2

First we use the contour $|z| = 3$. The map of $T(z)$ (Figure 8) shows there are three roots inside this contour, so we attempt to isolate one root. Figure 9 shows the result of using a square contour centered about the origin (cross) with a side of length 2; all three roots remain inside this square. Taking a still smaller contour, namely the circle $|z| = 1$, gives a single root; Figure 10 shows the contour, C, and the map $T(z)$. The first number represents the winding number.

$$N = \frac{1}{2\pi i} \oint_C \frac{dT}{T}$$

i.e., the number of roots inside the contour; it is equal to 1, within numerical accuracy. The second number is the position of the root,

$$R = \frac{1}{2\pi i} \oint_C z \frac{dT}{.T}$$

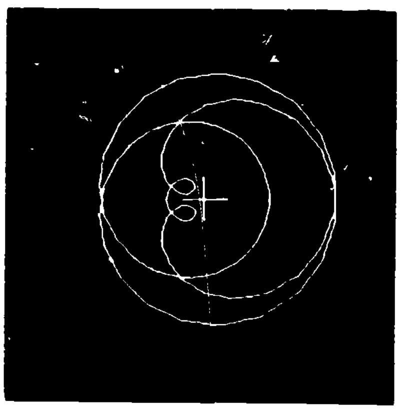


FIGURE 8 Map of circle $|z| = 3$ under transformation $T(z) = z^3 - z^2 \cos z + 3z + 1$.

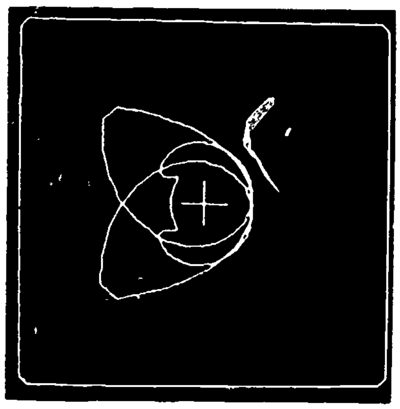


FIGURE 9 Map of square contour under $T(z)$.

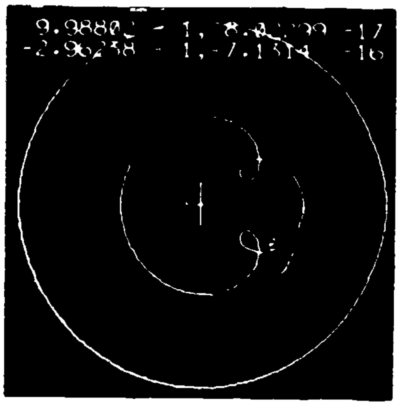


FIGURE 10 Map of unit circle under $T(z)$ displaying the values of the winding number and the enclosed root.

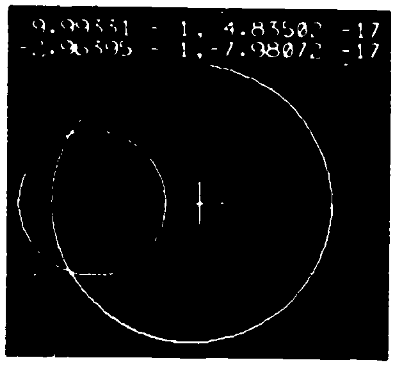


FIGURE 11 Map of circle of radius $|z - 0.3| = 0.2$ centered on approximate root of Figure 10.

This rough position is used as the center of a small contour to obtain a more precise value, as shown in Figure 11. Note that the root is real, $z \doteq -0.3$, and that its value has already converged to within 0.1%.

To search for another root, a 2 x 2 square is centered about $0 + i$. The result, in Figure 12, shows $N = 1.5$ roots inside the contour because it happens to pass through the root of Figure 10. Moving the contour up slightly (Figure 13) we obtain another root; again, using this as a center for a small circular contour yields the more precise value given in Figure 14, the change from Figure 13 to Figure 14 being less than 1%.

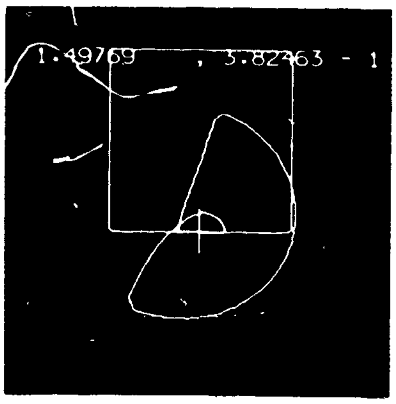


FIGURE 12 Map of 2 x 2 square centered at $z = i$.

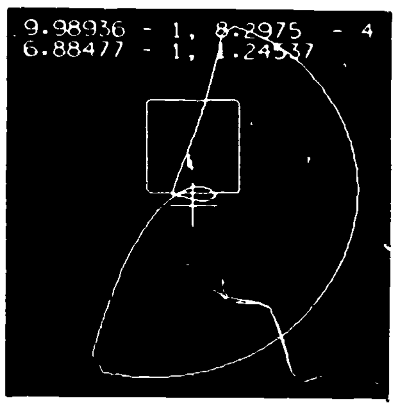


FIGURE 13 Result of raising the contour of Figure 12 slightly.

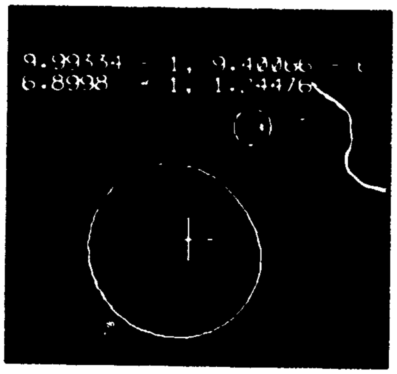


FIGURE 14 Map of circle of radius 0.2 centered on root of Figure 13.

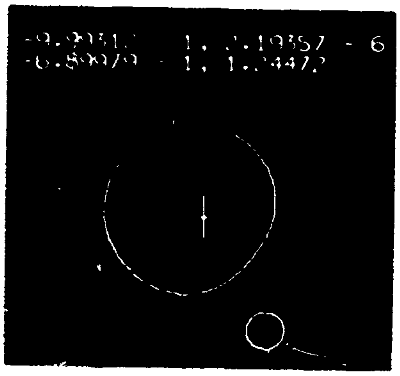


FIGURE 15 Result of using the complex conjugate of the contour of Figure 14 as the contour of integration.

Finally the third root is verified to be the complex conjugate of the second by simply conjugating the contour. The result (Figure 15) shows −1 root and gives the negative of its true position, because the integration is carried out clockwise due to the conjugation.

APPENDIX C: INFORMATION FOR PROSPECTIVE USERS

The on-line, mathematical system used for this work has been implemented on an IBM 360, Model 75 (at UCSB) and a Model 91 (UCLA). The programming is in assembly language. Typical user costs are of the order of $10-$15 per console hour, corresponding to about 1.5 minutes of CPU time on the Model 91. This average figure is about the same whether the user is a beginner or an experience practitioner.

It is entirely feasible for any users having an IBM 360, Model 50 or better, to run this on-line system, since the UCLA version has been prepared specifically for export, with special attention to compatibility with the standard IBM operating system. With one console, approximately 85K bytes or core suffice for the whole system; with five consoles, approximately 200K bytes are used. In addition, a console with two standard keyboards and a standard storage oscilloscope display is required, plus an interface between the consoles and the IBM 360. The total cost of this auxiliary hardware is about $20,000 to $25,000 for a one-console system, with each additional console costing $5,000 to $10,000, depending on manufacturer. Detailed information is available from Mr. William Kehl, Director, Campus Computing Network, University of California at Los Angeles.

REFERENCES

1. B. D. Fried, "Solving Mathematical Problems," in *On-Line Computing*, W. J. Karplus (ed.), McGraw-Hill, New York, 1967, pp. 131-175.
2. G. J. Culler, "User's Manual for an On-Line System," *op. cit.*, pp. 303-324:
3. *UCSB On-Line System User's Manual*, University of California, Santa Barbara, 1970.
4. R. E. DesMaisons, M. P. Hepner, and R. J. Dirkman, "THE BRAIN System," in this volume.

Inexpensive, Interactive Graphics on Telephone-Connected, Time-Shared Computers

JOHN R. MERRILL

This report presents a variety of (physics) illustrations of output from computer-connected graphic display devices, *XY* plotters and cathode-ray terminals, used in a relatively inexpensive, interactive mode of operation. The work has been carried out at Dartmouth College, under Project COEXIST, an NSF-sponsored project to investigate ways to use computers in introductory physics and mathematics teaching. This project is designed not only to improve the quality of the curricula, but to open up areas of study not previously available at introductory levels. Students use the computer in a number of ways on homework, on individual projects, and in the laboratory. Whenever possible students write their own programs; however, on occasion, it has proved useful to have programs written and stored in the computer by the instructor.

EQUIPMENT

The most important attribute of graphics systems is that, in general, the graphic output is usually more informative than tabulated numbers. While graphic displays do not have the resolution of tabulated numbers, they nevertheless convey a great deal more information in a short time. An *XY* plotting terminal of the type used extensively in Project COEXIST consists of a standard *XY* analog recorder with an interface device between the recorder and the computer. The interface converts digital characters coming from the computer over a telephone line into voltage positions on an *XY* plotter page. The computer software converts calculated values of *XY* coordinates to sets of ASCII* characters. The ASCII characters are the elements actually sent over the telephone line. Such an analog *XY* recorder is considerably faster in real time (and in computation time) than, for example, a CALCOMP incre-

---

Department of Physics, Dartmouth College, Hanover, New Hampshire.

*See *Glossary*, Appendix B.

mental plotter system. The cost of one of these inexpensive *XY* plotter systems is about $3300, of which the interface itself represents about $2000. A typical *XY* plotter system connected to a teletype and to an acoustic coupler is shown in Figure 1. A subroutine stored in the computer can be called from any program to position the *XY* pen anywhere on the page.

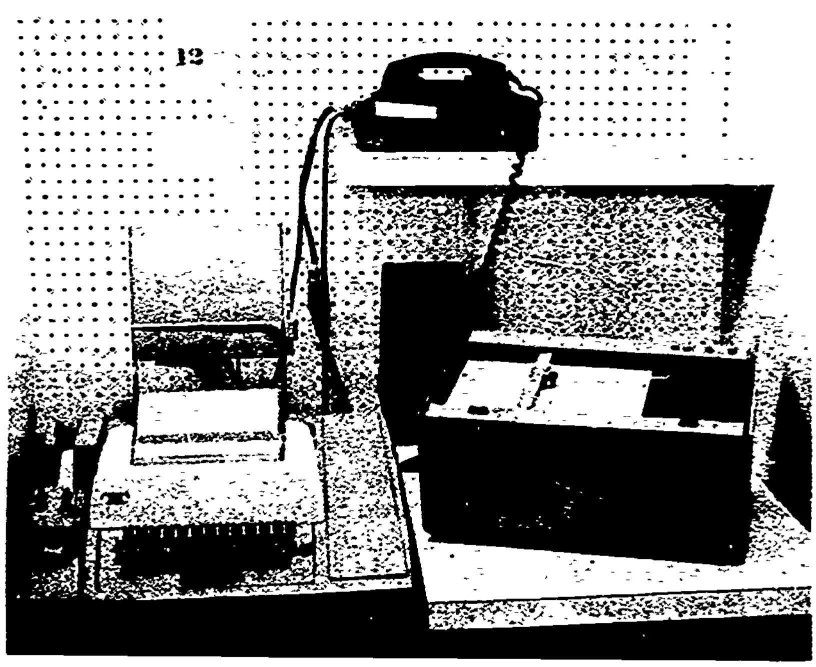


FIGURE 1 Teletype and *XY* plotter as used with the time-shared Dartmouth GE-635 system.

A second graphic display device is the Tektronix 4002 cathode-ray terminal (CRT) (Figure 2), a device that incorporated its own keyboard. Using the keyboard, letters are "printed" by the light spot on the screen. As with the *XY* plotter, a subroutine stored in the computer can be called from any program to position the light spot anywhere on the screen. Unlike the *XY* plotter, hard copy cannot be obtained from the cathode-ray terminal without photographic techniques or a hardware attachment. However, the cathode-ray terminal is often very much faster than an *XY* plotter, which makes the terminal useful for debugging and for those situations where the student must observe large numbers of curves. Since no inexpensive hard copy is available from the CRT display device, the CRT will not be emphasized. Nonetheless, all the examples in this paper have been executed on the CRT device.

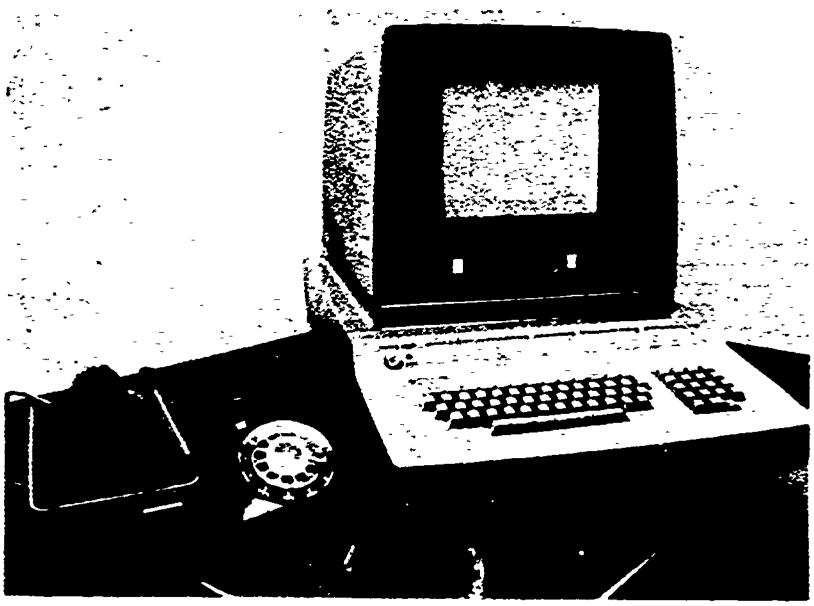


FIGURE 2 Cathode-ray display terminal as used with the time-shared Dartmouth GE-635 system.

ILLUSTRATIONS

Illustrative problems from classical mechanics, fluid flow, electrostatic fields, geometrical optics, and physical optics have been chosen to demonstrate the ways in which graphic display devices have been used. In classical mechanics, students have written and used programs in such areas as trajectory motion, strobe photograph labs,[1] Keplerian and non-Keplerian orbits, relativistic dynamics (including relativistic motion of charged particles), accelerator simulation, and a number of scattering situations. The Figures 3-6 are results of one sophomore-level scattering simulation laboratory.

Particle Dynamics

Figure 3 shows the results of the classical scattering of point positrons from a model of an *S*-state hydrogen atom. The atom is modeled as a point nucleus surrounded by a uniform negatively charged sphere of unit radius. The total negative charge in the sphere exactly cancels the total charge of the point nucleus. The scattered particle is repelled from the nucleus. The figure shows the trajectories of positrons with an energy of 25% of the ionization energy (13.7 electron volts) and for various impact parameters. The scattering is

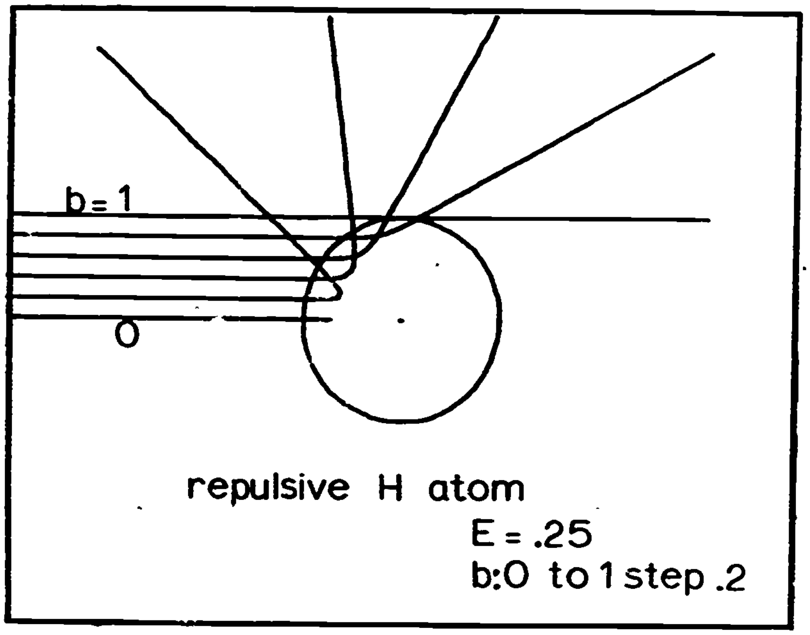


FIGURE 3 Classical scattering of a point positron from a model of an *S*-state hydrogen atom. Energy is in units of 13.7 eV, b is the impact parameter.

entirely classical. The student becomes familiar with the concepts of impact parameter, angular momentum, differential cross section, total cross section, and effective potential in a classical system. He then finds these concepts much easier to understand in quantum-mechanical cases.

Figures such as Figure 3 are used in a computer laboratory simulation experiment. The student starts with a hard sphere model. After plotting and understanding the trajectories of hard sphere scattering, he then plots trajectories such as those shown in the figure. He measures the angle of deviation and plots the number of particles scattered into 20° intervals of scattering angle as a function of the angle of deviation. The plot is normalized in such a way as to make it a plot of differential cross section. The student gets results like those shown in Figure 4. This figure shows the number of particles scattered into a given angle as a function of scattering angle. Notice the large number of small-angle scatterings. Since the potential is cut off by the electron screening, the number of small-angle scatterings is large but not infinite.

The scattering angles are grouped in 20° intervals, so the differential cross section is then the number in an interval times the impact parameter b, divided by sin $\theta(\Delta\theta)$. The student gets even better data than that shown by using smaller interval sizes and more trajectories. The total cross section in

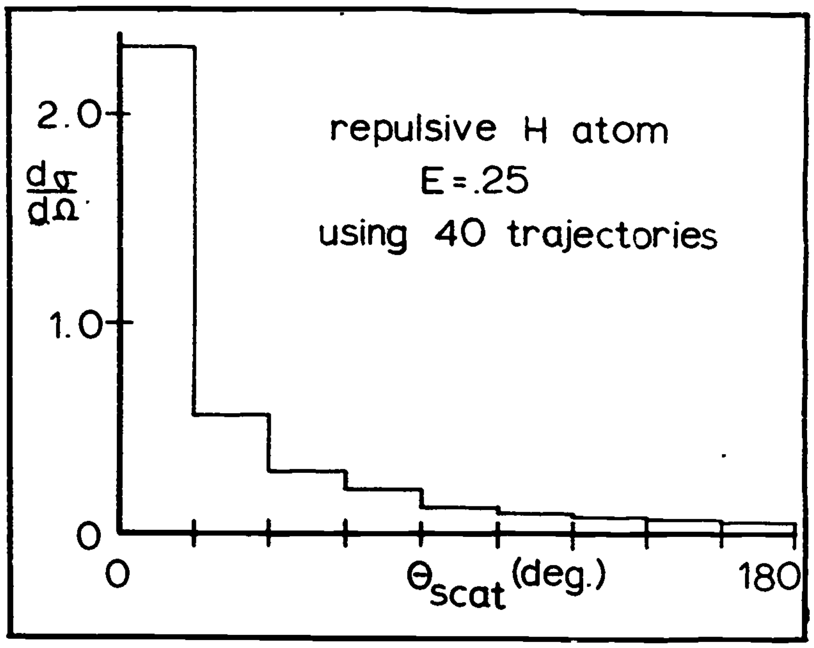


FIGURE 4 Differential cross section as a function of scattering angle for trajectories such as those in Figure 3.

the normalized units used in these plots should be approximately π since the normalized radius of the model atom is 1, and in general the total cross section should be πR^2. If the student adds up the results for the differential cross section, he gets a total cross section typically between 3.12 and 3.15. The data from this figure give a value of 3.12.

Figure 5 shows this same *S*-state hydrogen model for electron scattering. That is, the scattered particles are attracted to the central nucleus. Notice the peculiar looping orbits. These orbits are correct, and the student must explain these effects. The answer is seen, most easily, in the effective potential, V_{eff}. Figure 6 shows the effective potential for various impact parameters. The physical situation may be understood with the aid of Figure 5. Near an impact parameter of 0.9 (for normalized energy of 0.25), the negative charge just surmounts the peak in the potential. Since the square of the radial velocity is proportional to the total energy minus the effective potential, the particle slows down radically while passing over the peak in V_{eff}. Conservation of angular momentum demands that the charge wind around the nucleus a number of times. Ultimately the electron leaves the atom, but for angular momenta very near 0.9 it may take an arbitrarily long time to get away.

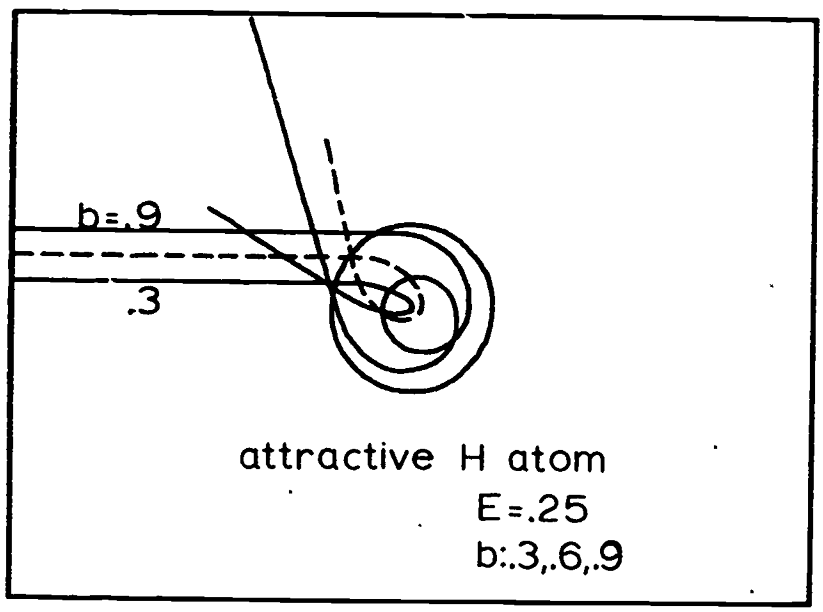


FIGURE 5 Classical scattering of a point electron from the model of an *S*-state hydrogen atom.

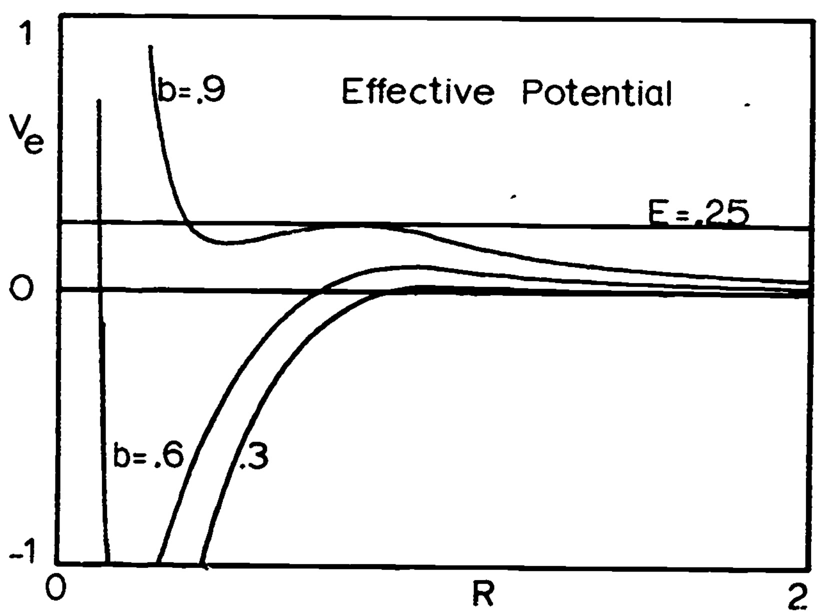


FIGURE 6 Effective potential for several impact parameters for electron scattering from the model of an *S*-state hydrogen atom. The curves correspond to the trajectories shown in Figure 5.

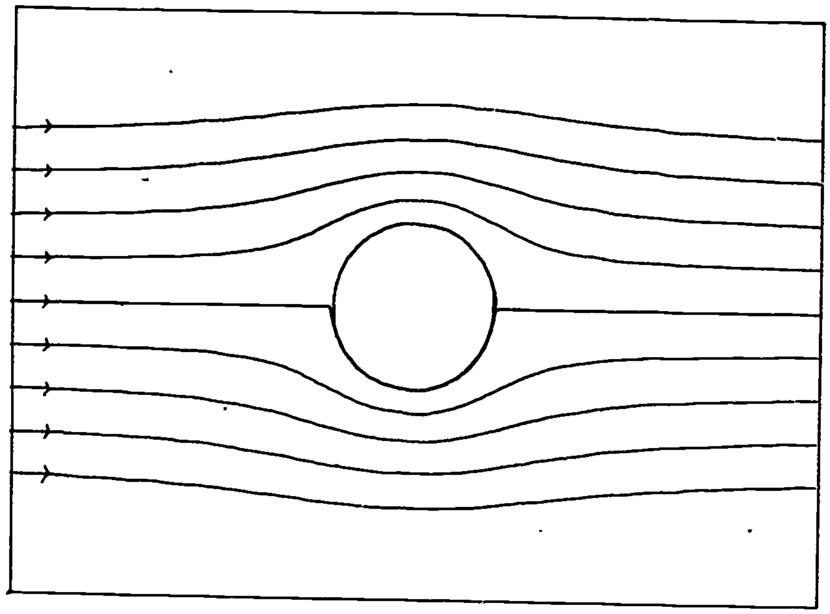


FIGURE 7 Velocity flow lines for fluid flow around a cylindrical object.

Field Plotting

Graphic displays have been very useful with various field patterns. The concept of vector fields has been introduced by means of flow patterns in hydrodynamics. Figure 7 shows the pattern of the velocity field around a cylindrical object placed in a uniform stream. A student plots such flow patterns for a number of objects. Then he finds densities and directions of lines in order to derive the relative velocities at different points in the pattern. In this way the student not only acquires some information about fluid flow but also about various ways to represent vector fields. Vortex (rotational) fields in fluid flow can often motivate discussions of circulation leading to the concept of the curl of a vector.

Figure 8 shows the plot of the electric field lines and equipotentials around a two-dimensional quadrupole. This program allows placing point charges anywhere in the plane. The program then follows the field lines by integrating $dx/E_x = dy/E_y$. The program follows the equipotential by a hunting routine which follows the equipotential contour. Programs have also been developed which follow the equipotential contours by moving everywhere perpendicularly to the local electric field lines. These programs and plots are useful in that the student very quickly gains an intuitive grasp of the meaning of the abstract concepts of field lines and equipotential surfaces.

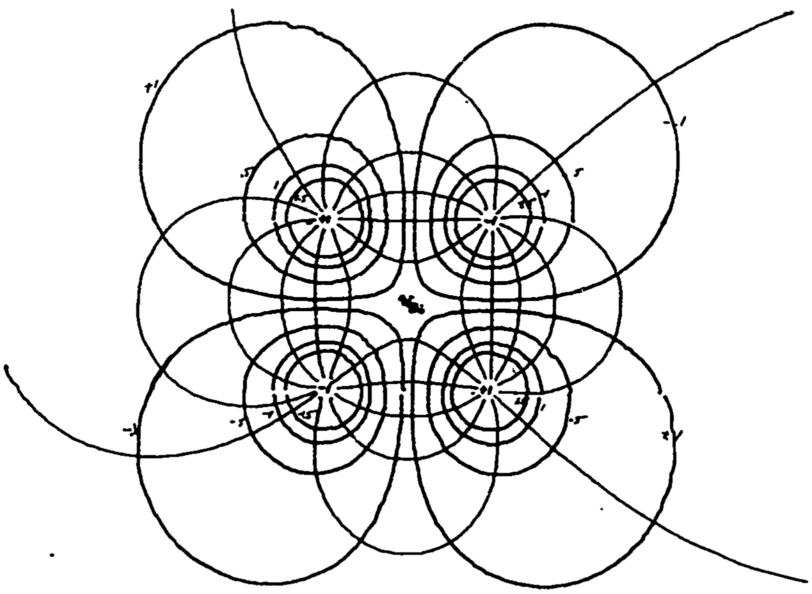


FIGURE 8 Electric field lines and equipotential surfaces in the plane of a two-dimensional quadrupole.

Optics

In geometrical optics we have developed programs to illustrate the tracing of principal rays through a thin lens or spherical mirror optical system. By plotting rays for several systems the student understands imaging much more quickly. True ray-tracing programs have also been developed in which the student can place any number of spherical interfaces between media anywhere in the plane. The student then starts the ray at some angle and at some position, and the program traces the ray through the system. This program demonstrates nicely the various forms of aberration, as shown in Figure 9. The shift in focal point due to spherical aberration is apparent.

Figure 10 shows another application of the computer to introductory geometrical optics—mirages. The program allows the index of refraction of the medium to be a function of height. The figure shows the results for a model index of refraction near heated ground. The sheet of less dense air near the ground produces a second, inverted, virtual image of any object above the surface. One observes two objects—one at the true positions of a tree; the second a mirage image. This program can also be used to demonstrate "looming." Looming is an upside-down mirage due to a layer of warm air sandwiched between layers of cold air, which probably explains most sightings of

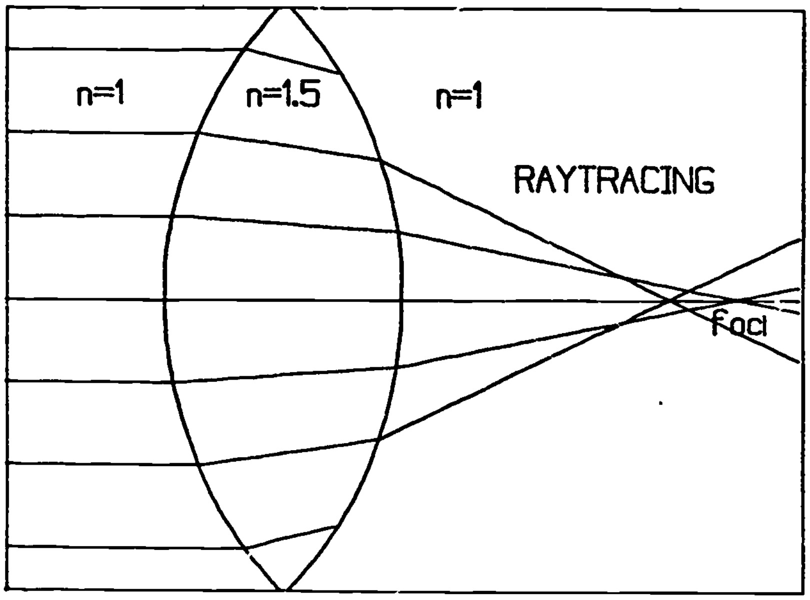


FIGURE 9 Rays traced through a thick lens.

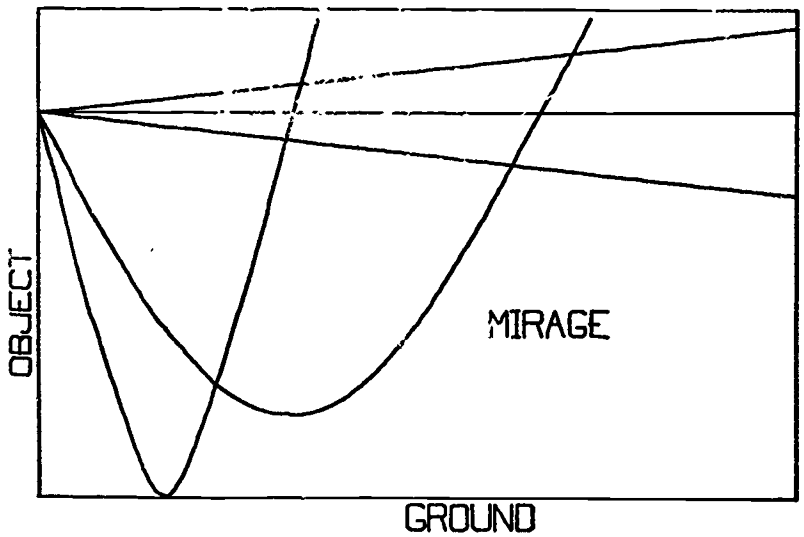


FIGURE 10 Mirage effects when the index of refraction varies with altitude.

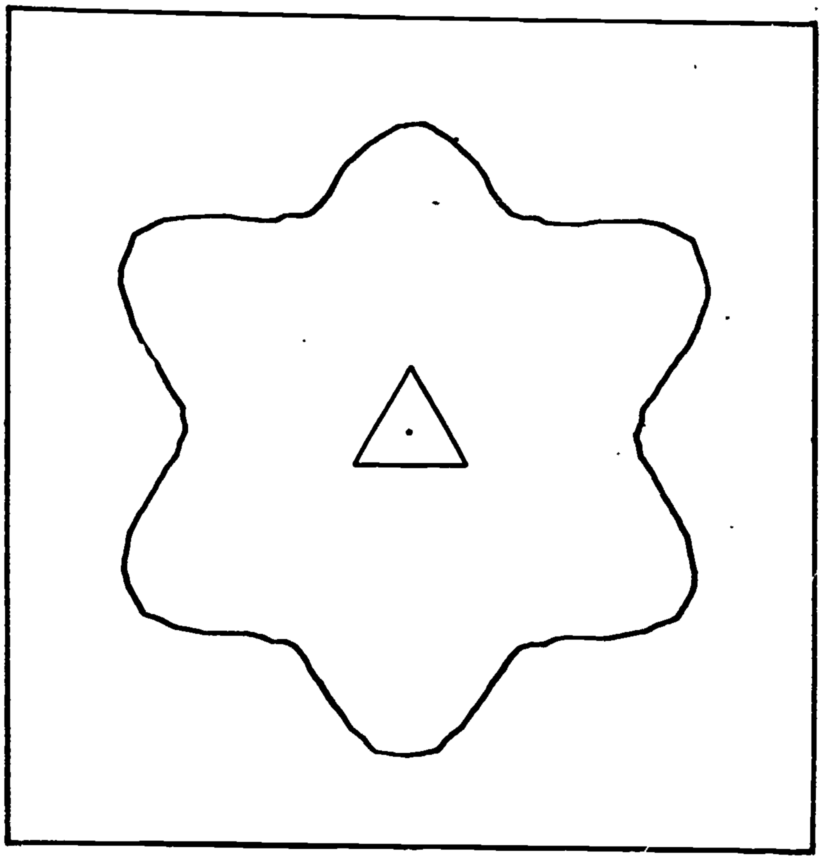


FIGURE 11 Intensity of emitted radiation versus angle from three synchronous sources placed at the corners of an equilateral triangle of side $\lambda/2$.

the "Flying Dutchman." The program also illustrates reflections of radio waves off the ionospheric *F*-layer.

In physical optics, we have programmed a number of applications of Huygen's Principle. Such programs, in addition to use in showing the production of *N*-slit diffraction patterns, have also led to more complex applications. Students have dealt with arrays of radio antennas placed to maximize the directionality for transmission or reception. They have studied broadside arrays and interferometric radio telescope arrays. In these programs the student compares plots of intensity versus angle and intensity versus position. Figure 11 is a plot of intensity versus angle for three line sources placed on a equilateral triangle of side length = 1/2 wavelength.

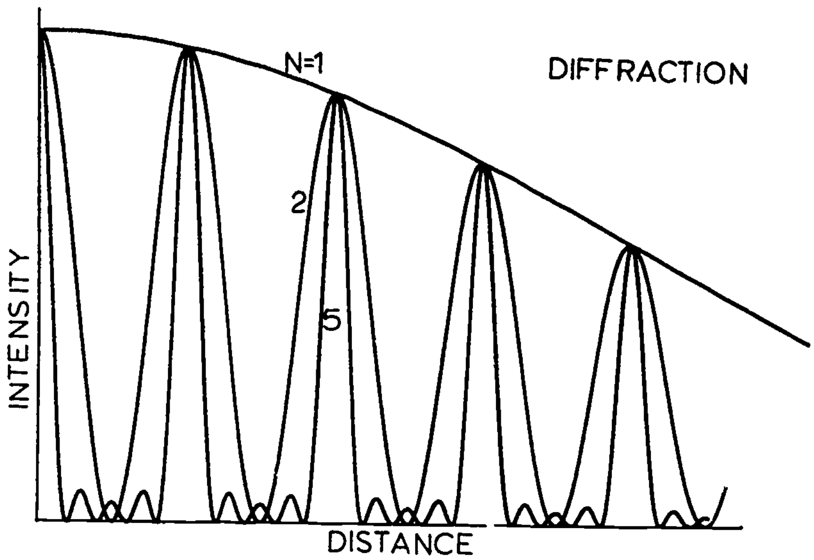


FIGURE 12 1-, 2-, 5- and 10-slit diffraction patterns for data from a student laboratory.

A laboratory application of the computer in introductory physical optics is shown in Figure 12. This shows (normalized) *N*-slit diffraction patterns from 1, 2, 5 and 10 slits using the geometry and wavelength given for an introductory lab. The student measures the intensity versus position across the observation screen of an *N*-slit experiment using a photoresistor and a helium-neon laser. He plots his results superimposed on a theoretical plot such as Figure 12; agreement is usually good.

SUMMARY

This paper has illustrated a number of uses for graphic display devices in introductory physics teaching. All the illustrations used inexpensive *XY* plotter systems in communication with the computer. The programs have also been performed on a CRT terminal. Some of the plots also have been performed on the teletype itself using conventional printout. It is hoped that a large number of examples of the use of graphic display devices illustrates the importance of these devices better than volumes of words, and perhaps these examples will inspire other applications in the reader's mind. A glance at a plot often leads to more understanding than a long look at a series of numbers in tabular form.

A short comparison of the three methods of plotting we have used might be helpful. The CRT is faster than either the *XY* system or the teletype, and it is useful for debugging or for situations where large numbers of curves must be displayed quickly. The CRT is essentially silent when compared to a teletype, but also relatively expensive when compared to a teletype and an *XY* plotter. The *XY* recorder plotting system is much faster, in general, than a teletype and has much higher resolution; it automatically gives the student a hard copy of his results for further deduction. It could well become the workhorse of introductory teaching applications. The plots are easy to handle, scale, and even label if the student wishes. The teletype itself can be used to plot some curves; often however, the resolution available is insufficient to show anything but coarse behavior. Nonetheless, teletype plotting is a useful way to introduce students to plotting in general and can be very useful in debugging plotting routines fairly quickly. Furthermore, higher resolution printing heads and software designed to employ them for more accurate plotting on teletypes are just now becoming available.

REFERENCES

1. E. R. Huggins, "A Strobe-Computer Laboratory for Introductory Physics," in this volume.

On-Line Classroom Graphic Display

TIM G. KELLEY

The commercial development of moderately priced computer-driven graphic terminals has at long last opened the door to interactive computer usage right in the classroom. The capability of in-class graphic display of information generated on-line has long held very great appeal as an instructional concept,[1] since the most difficult task facing the science teacher is in the translation of mathematical equations and symbols into mental constructs and graphical images that have physical content and meaning to the student. Unfortunately, without very specialized and expensive hardware the dual requirements of a time-sharing system and special digital-to-analog conversion equipment have previously not allowed the practical realization of classroom use of computer-driven graphic display systems.

However, it is a long and difficult step from hardware availability to viable and sufficiently flexible classroom use. It is the purpose of this paper to discuss some of the requirements which the driving software systems must meet in order to be generally useful in the classroom, and to describe our attempts to meet these requirements in a pilot project in physics at Oregon State University.

SOFTWARE REQUIREMENTS

First and foremost, the programs for interactive graphic display systems must contain the highest possible degree of flexibility. To be interactive, *the instructor must be able to alter dynamically the display of a problem's solution, or set of solutions, for emphasis, clarity, or in response to direct questions.* Since one of the principal purposes of graphic displays of information through any medium is to reveal the effects of parameter variation, a fairly flexible parameterization scheme must be incorporated so that meaningful families of solutions can be generated and graphically displayed, either sequentially or simultaneously.

Most classroom topics are many-sided in that there may be several quite

---

Department of Physics, Oregon State University, Corvallis, Oregon. This project was supported in part by NSF Grant GJ-51, NSF Aufenkamp 370 Regional Center.

different aspects to the solution of a problem. For example, consider a central force orbital problem. Upon a given day the pertinent display could be any or all of the following: plots of independent variables (r,θ) vs. time t; plot of $r(\theta)$; plots of $r(\theta)$ with real-time development; segmented plots of $r(\theta)$ to demonstrate the equal areas concept or to indicate graphically the calculational method by which a computerized approximation to the solution to differential equation systems is made[2,3]; plots illustrating the energetics; displays of the effects of perturbations and comparisons of perturbational calculations with "exact" solutions; etc. The list is possibly endless. Clearly, an essential requirement, then, is a complete diversity of available output options. Here, again, *the advantage of on-line access over more conventional media, such as movies, is that the demonstration can be molded to fit the lecture and not vice versa.*

Of nearly equal importance with flexibility is the requirement of "viability." This requirement can be divided into at least two main categories. First, the input required must be simple enough that very little of the lecturer's attention has to be directed to it. Second, the form of input must be such that even an instructor unskilled in computer techniques can learn to use it with confidence in a time no longer than would be required to set up any other form of lecture demonstration.

Among the most important aspects of these requirements are:

(1) convenient and efficient procedures for identifying and specifying input data, incorporating a very free form of input, and with a simple means for varying input parameters to obtain comparative families of curves;
(2) a minimal method for changing parameters;
(3) flexible and convenient procedures for specifying output. In addition to the various aspects of the physical problem itself which may be pertinent, there should be options such as sequential or simultaneous display of solutions for different parameter values, "zooming" in or out (on a time series, for example), display of digital information, requests for hard copy of the plots or solutions, etc.;
(4) effective error recovery techniques. Errors are inevitable, and valuable class time cannot be wasted when they occur;
(5) timing control mechanisms.

A principal advantage of on-line computer displays over movies, for example, is the potential capability for the instructor to dynamically control the timing and sequential structure of the demonstration. To accomplish this in a flexible way without requiring a great deal of his attention we have developed an input/output "language." This currently consists of twenty "tekplot commands" or "tekcommands," most of which trigger the plotting of another solution as well as provide selectivity. The full complement of twenty commands and their optional modifiers provides an instructor with a highly flexible means of controlling the presentation; however, an instructor with no patience for computer languages can get by quite nicely with only the two simplest:

Carriage Return—trigger the next curve of a family.

STOP—terminate this series of plots.

(The commands have names like "STOP" and "CLEAR" for mnemonic purposes, but with the present array of tekcommands, at least, it is only necessary to input the first letter.)

A PILOT PROJECT

To study the development of classroom computer-driven displays we selected a pilot project—a pair of linear oscillators, arbitrarily coupled, damped, and harmonically driven. Although a simple problem mathematically, its input/output requirements are typically as complex as any. A fortunate feature of this choice is that it contains an even simpler but also useful subproblem, that of a single oscillator, which provides a convenient vehicle through which to initiate other instructors in the operation of the terminal. It will serve nicely, here, to illustrate our methods in general and to provide an overview of the nature of the tekplot commands before presenting a more detailed discussion of their meaning.

Figures 1 through 10 comprise a sequence of photographs of the terminal screen taken during a practice demonstration. In the first two pictures the actual input required from the instructor has been underlined, with the rest of the text being displayed by the computer (at a *very* rapid rate). Figure 1

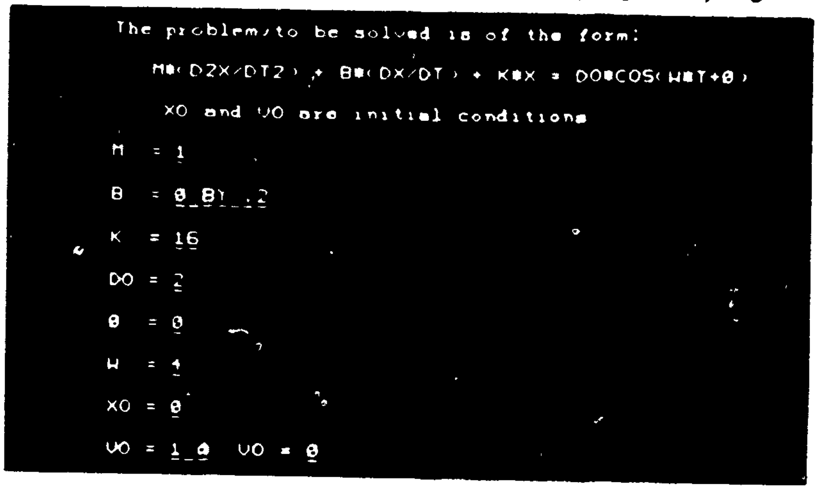


FIGURE 1 Initial specification of oscillator problem.

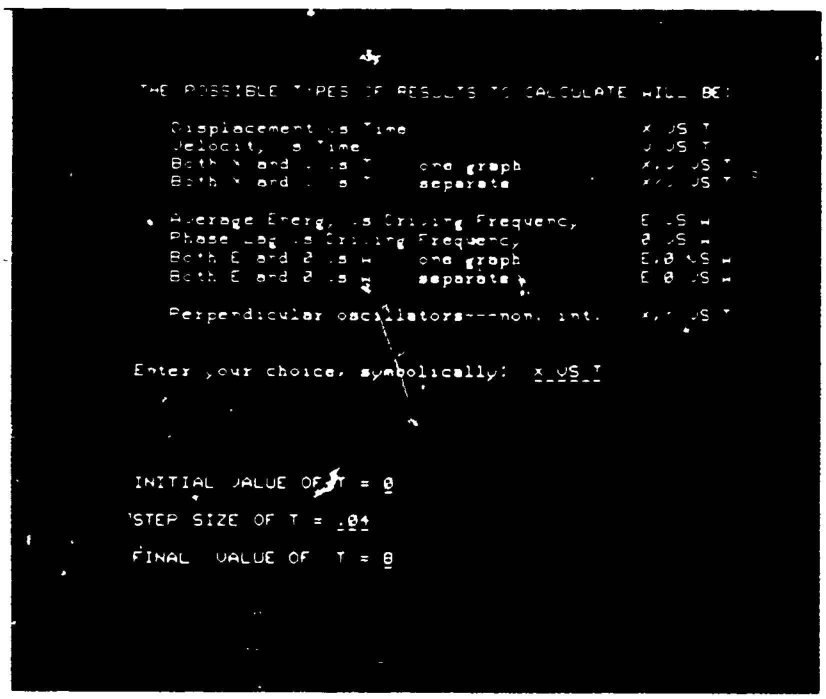


FIGURE 2 Selection of computational options.

illustrates the input of parameter values and shows how a parameter (B) can be specified as automatically varying (in this case in increments of 0.2). An editing capability for error recovery is associated with all input, such as the use of '@' in the input for VO, here. This caused the existing input line for VO to be canceled and a new value requested. The same recovery could have been achieved by typing a backslash, (\), which would have canceled only the previous character. Other error recovery facilities have been incorporated, such as provisions for jumping back to the start of an input sequence, for example. In Figure 2 the selection of the type of output results is made and the initial range and step-size of the independent variable, in this case *T*, are specified. These, as well as any of the parameters can easily be changed later if any turn out to be poor or mistaken choices.

The computer normally does everything possible to determine appropriate graph limits without bothering the instructor (who can, however, supersede

FIGURE 3 Pure resonant amplification, X = T*SIN(4T).

the computer with the LIMITS command), but in this case the first solution is unbounded, necessitating one further inquiry (not shown). Depressing the carriage return (CR) then produced the axes and their labels and the oscillating curve of Figure 3. The positions of the arrows were quickly specified through the graphic input "joystick" device, as was the drawing of the envelope shown. The curve label, B = .00 WILD, is automatically provided, but the other three messages were typed manually. The envelope could also have been obtained by typing "X=" followed by its equation, if it were known.

Successive curves in the family, determined by the value of the varying parameter B, could be triggered one at a time by depressing (CR). In this case, Figure 4 was generated by preceding (CR) with the modifier '3' which caused the triggering requirement to be suppressed for the 3 curves. Instructionally, Figure 4 shows clearly the expected result that response to a disturbance is inhibited by damping effects, but does not yet suggest the

FIGURE 4 Resonant amplification curves for damping constant B = 0.0, 0.2, 0.4, and 0.6. In the last instance, energy input exactly balances resistive losses, oscillation is steady.

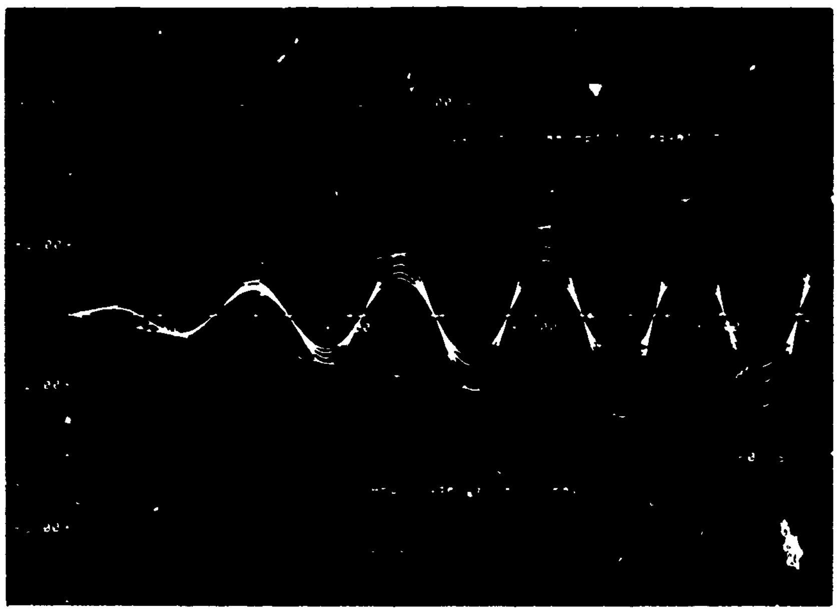


FIGURE 5 Extension of graph of Figure 4.

existence of a steady state. Giving the EXTEND command (Figure 4) caused the subsequent time segment shown in Figure 5 to be generated, the modifier '4' again suppressing the triggering requirement. The TOTAL command then caused the *concatenation* of the curves in Figures 4 and 5, producing Figure 6.

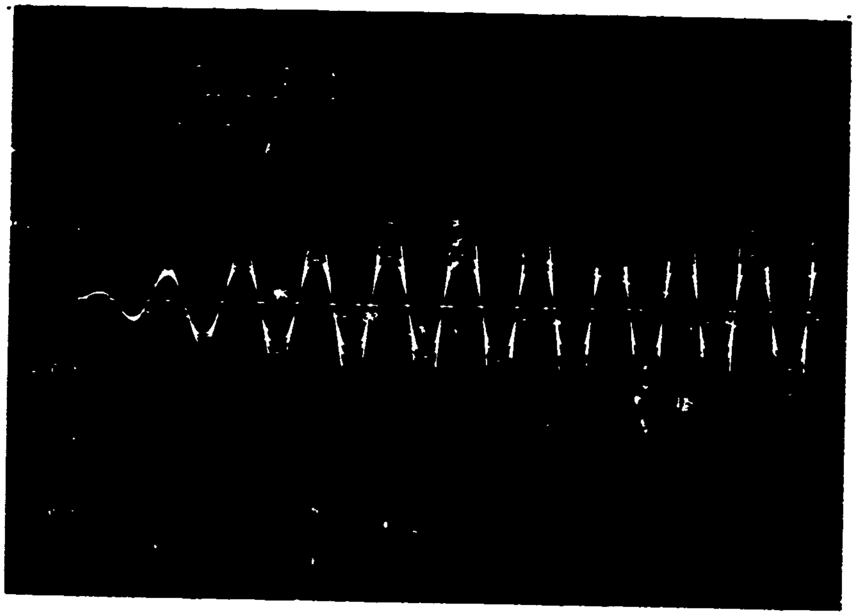


FIGURE 6 Concatenation of Figures 4 and 5.

The option of graphically examining relationships "in the small" before trying to assimilate the "big picture" offers a decided advantage over textbook illustrations which often attempt to condense as much information as possible into the smallest amount of space. Figures 7 and 8 illustrate another method of sequentially drawing the student's attention to small portions of the display. First superseding the normal assignment of plotting range for the independent variable through the LIMITS command (not shown), the successive segments (8-16 and 16-24) of Figure 8 were then triggered for both dependent variables by depressing the line feed key. Yet another method allowing examination of detail is demonstrated in Figures 9 and 10. Here a small portion of an existing display is magnified through the ZOOM command. The boundaries of the region to be magnified can be specified either through modifiers appended to the command or via graphic input. Thus the positioning of the "cross-hairs" shown in Figure 9 determines the boundaries of Figure 10, the scale of the graph being expanded accordingly.

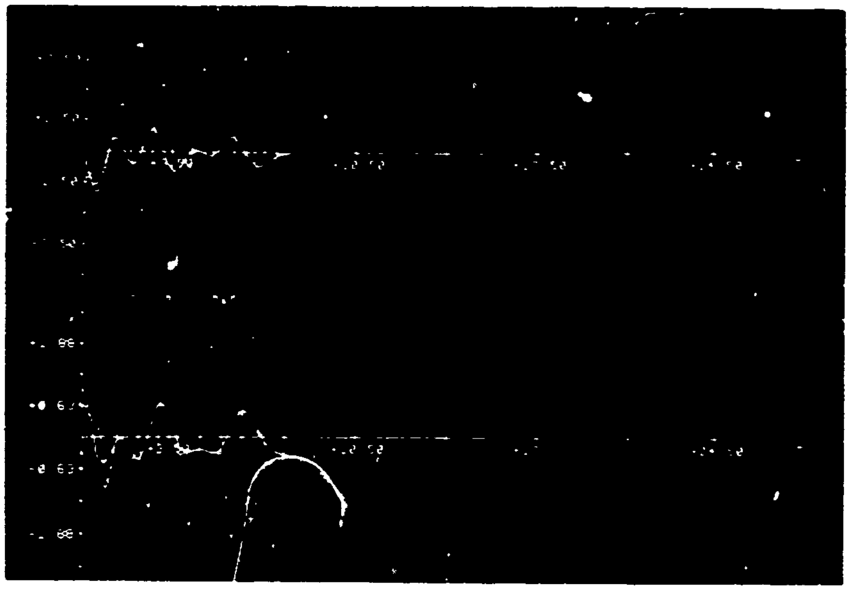


FIGURE 7 Graph of initial stage of oscillation.

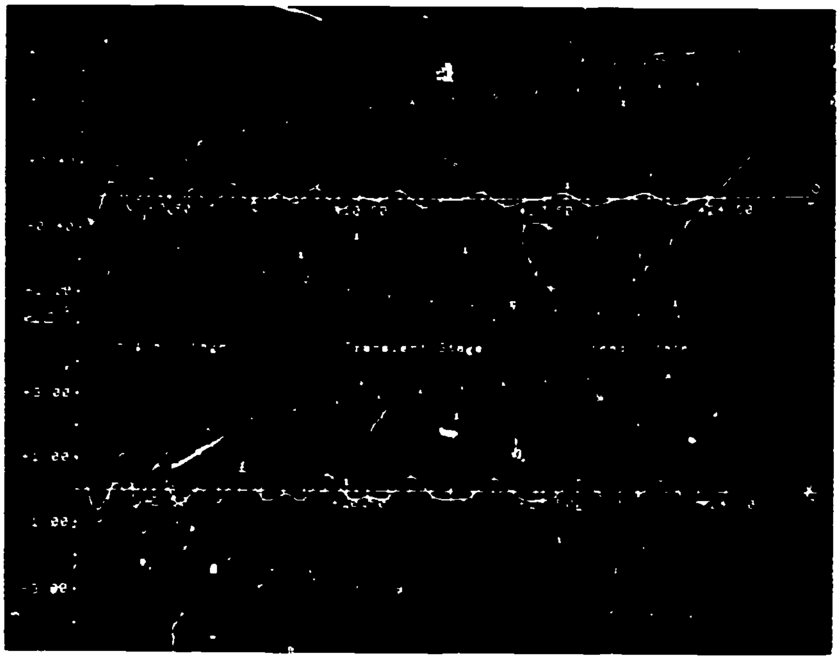


FIGURE 8 Completion of Figure 7 by addition of transient and steady-state behavior.

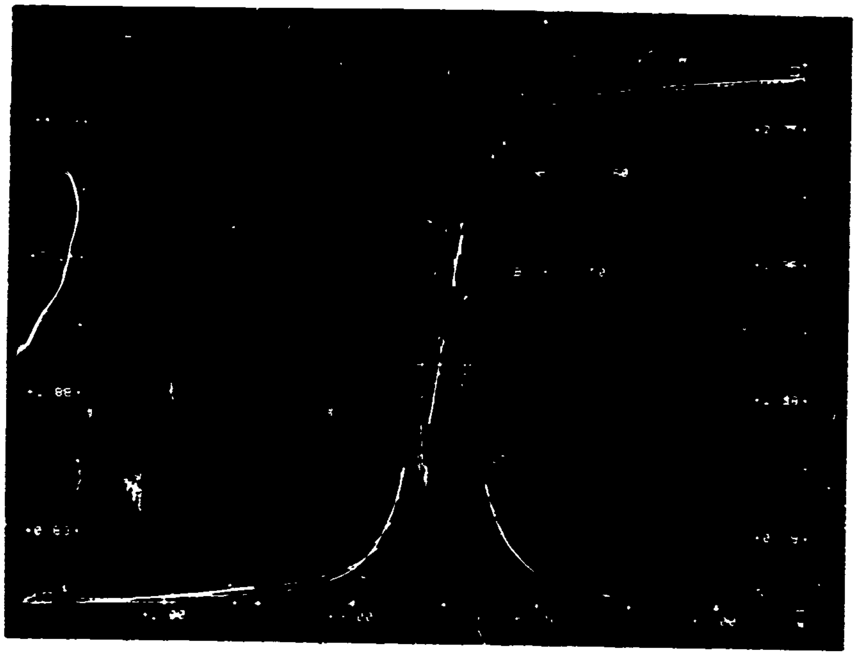


FIGURE 9 Amplitude and phase-shift as a function of frequency before magnification by ZOOM.

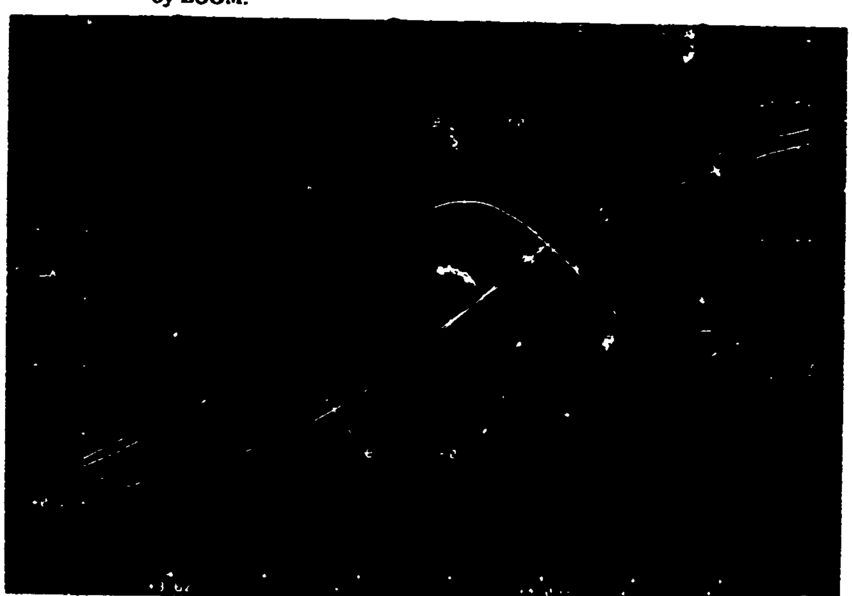


FIGURE 10 Magnified view of upper right quadrant of Figure 9 as specified by position of "cross-hairs."

THE TEKCOMMANDS

Although an instructor can make do with only two or three tekplot commands, a far more versatile display capability can be acquired by increasing his vocabulary to encompass a greater portion of the available commands. We now present a brief description of these commands.

Think of trying to display a family of curves, each specified by a set of values for the parameters specifying the problem. These parameter value sets can all be prespecified as part of the initial input, or, alternatively, only those for a particular curve could be given along with their corresponding increments. The first method is definitely the clumsier and requires more of the instructor's time but could be quite appropriate for specifying initial conditions for a set of coupled differential equations, for example, where the "family" might consist of a curve for each dependent variable. If the method of increments is chosen, then it may often be desirable to change the *increment* values at some time during the demonstration; this is accomplished through the tek command INCREMENT.

In most cases the presentation of an entire family of curves at once (as in textbooks) is overwhelming and much less effective than showing and discussing them one or two at a time. Thus a means of "triggering" successive curves is required. It is clearly preferable that this be a "dark" command, i.e., one leaving no writing on the screen, and hence the carriage return key (CR) was selected for this purpose to be the "trigger." It may be, for some reason, that some of the curves are better skipped than shown; depressing the space bar *n* times before CR will accomplish this by causing *n* curves to be skipped over before the next one is plotted. Additionally, typing an integer *m* just before CR causes *m* successive curves of the family to be plotted. These same results can also be obtained through the SKIP, *n*, *m* command, except that it is not dark; but if *n* is large this may be a more convenient form. In principle this process could continue indefinitely, the screen becoming totally cluttered with curves that are no longer of interest. Hence the command CLEAR, *n*, *m* which produces the same effects as SKIP, but the screen is erased and the axis redrawn before the new curves are plotted, in effect initiating a new family of curves.

Thus the three commands, CR, SKIP, and CLEAR, along with INCREMENT, comprise a simple set of instructions for flexibly controlling the timing and the selection in displaying a family of curves. This selectivity is completed by the option available at any time to arbitrarily assign a new value to any of the defined parameters by typing a simple replacement statement preceded by '↑' such as ↑D = 2.

Let us now consider the kinds of control over the independent variable necessary to provide a sufficiently flexible operation. Whether the independent variable of the problem is one of the coordinates for the displayed plot, it is the range of this variable that determines the detail the displayed curves will portray. Thus it may be advantageous to present some families of curves in "segments" in order to support a discussion of the detail. A mechanism is then necessary for shifting to the next segment, and this is accomplished through the command, EXTEND, *n, m* (the optional modifiers *n* and *m*, here as everywhere, refer to curve specifications and have the same meaning as for SKIP). After viewing a number of such consecutive segments and digesting all the detail, it will usually be desirable to place this in proper context by combining all the segments into a single figure. This is the action of the TOTAL, *n,m,i,f* command, with the optional modifiers *i* and *f* specifying the range of the independent variable, from "initial" to "final" values.

It is also possible, and may often be desirable, to reverse this process, i.e., to present the single figure first, and then to scrutinize for detail. Exercising the PLOT command allows a totally arbitrary relocation with respect to the independent variable, as well as a respecification of the segment size. (The segment size may also be redefined at any time by means of the command SEGMENT.) The PLOT command serves another function in that it makes it possible to skip over uninteresting or repetitive portions in the domain of the independent variable.

Although these commands provide control of the *visual* detail, none of them causes any enhancement of the *computational* detail, i.e., the basic step-size of the independent variable never changes with any of these instructions. A true magnification of detail is afforded through the ZOOM command, which preserves the *number* of steps in the displayed curve, not the step-size (see Figures 9 and 10).

In addition to these controls over the independent variable, there is a special-purpose command, (LF), which consists simply of depressing the line feed key, preceded possibly by striking the space bar a few times. Formally this command accomplishes the same thing as EXTEND does, except the screen is not erased before the next segment is plotted! This clearly only makes sense if the next segment does not overflow the screen, which could be the case, for example, for multiple-valued curves, such as in an orbital problem, or in a diagrammatic construction. If the independent variable is a plotting coordinate, it is possible to previously force the plotting range to cover several consecutive segments through the LIMITS command (see Figures 7 and 8).

To review: The commands EXTEND and TOTAL, along with PLOT and SEGMENT provide control over visual detail. The ZOOM command offers computational magnification, and the special "dark" LF command allows a timing control over the independent variable somewhat analogous to the timing control provided by the CR instruction in parameter space. The remaining tekplot commands are of a more auxiliary nature and will not be discussed here. Actually, most of their names make them almost self-explanatory, anyway. With the exception of STOP, they will be used infrequently, and some are not even implemented yet.

BAD PLOT	FILE DATA	HARD COPY	LIMITS	STOP
DUMP	GET DATA	INFO	REAL TIME	VALUES

It is anticipated (hoped?) that the above-described display terminal software will find application to a wide variety of instructional problems, for some of which the generation of data may be quite expensive; a data storing system has naturally been devised to avoid duplication. The storage is essentially "bottomless" in that peripheral devices (disk, tape, etc.) are automatically utilized whenever core allocations are exceeded. Because it is possible to generate quite disjointed data through the use of the tek-commands, an unfortunately complex set of pointers and tables is required to support this data base. The problem is further complicated by the need to minimize core storage requirements, which for ours and many other time-sharing systems is a critical factor in determining "response time."

A coordinate on the display terminal screen contains about 10 bits of information; a prudent storage scheme clearly involves some packing of the data. Finding solutions to these and other related problems has been a very time-consuming and tedious task which, it seems, will have to be duplicated by anyone attempting to develop a similar system. For that reason we have prepared a very detailed description and set of flow charts for the data-base system, copies of which will be made available to anyone upon request.

SOME FINAL REMARKS

An extremely important consideration is the modular structure of the software itself. If the capability of the described system were limited to displaying simple harmonic oscillator solutions or even coupled oscillator solutions, we would quite literally be employing the legendary sledgehammer on the proverbial peanut. Actually, however, the tekcommand executive routines, the data storage and retrieval routines, and the conversational I/O routines are all designed to operate on abstract variables and arrays which could pertain to any problem. These routines simply make requests to a problem-dependent "data generator" to obtain whatever data they need to carry out their mission. Thus the generation of demonstrations for new problems requires only the programming of the mathematical solution(s) to the problem, following a few uncomplicated rules for interfacing it with the described display terminal software, and loading a few files or arrays with appropriate labels and messages.

Born of our experience with the coupled oscillators, the development has nevertheless been strongly guided by the anticipated requirements of other, quite different types of problem. A re-examination of the previous section will show, for example, that any of the displays mentioned earlier in conjunction with central force problems can readily be generated through the tekcommand structure. Recently we have implemented and interfaced a powerfully versatile data generator to the display software in the nature of a formula compiler. Utilizing this, the instructor need only type, using ordinary mathematical terminology, the mathematical expressions of the quantities he wishes plotted. Parameters are defined by their occurrence in these expressions; their values are then requested by the computer exactly as in Figure 1.

Although the project has not yet reached the stage for classroom and student testing, several instructors with no computer experience have gone through the process of learning to use the single harmonic oscillator demonstration with very satisfactory results. Spending 20-30 minutes reading a prepared writeup and an hour or so at the terminal typically suffices to achieve a level of competence adequate for classroom presentation, although the full language cannot be learned in that time.

The figures bring out a very important point in this regard. Since the graphic computer terminal writes at such a very high speed, it is quite feasible to instruct or remind the user how to run the program *as he goes along*. This not only facilitates the process of learning to use the graphic displays, it should also serve to dispel some of the inborn distrust and reticence many teachers seem to feel for computer usage.

I wish to acknowledge the cooperation and assistance of the OSU Computer Center staff, in particular Mrs. J. Baughman and Mr. G. Rose. Also helpful were the criticisms of C. Fairchild, D. Griffiths, N. Wise, R. Craig, W. Au and L. Hogan of the OSU Department of Physics, G. Wolfe of Southern Oregon College, Paul Lineau and Paul Chitwood of the Oregon Technical Institute, and B. Cantrell of Portland State University. I am greatly indebted to Mr. Mark Ebersole, a physics student at OSU who is responsible for most of the programming and other valuable contributions to the project.

We are grateful to Tektronix, Inc., for the loan of the pilot model of the Graphic Computer Terminal (T-4002) and accompanying equipment, and to Mr. R. Hornicak of that company for his cooperative assistance.

APPENDIX A: INFORMATION FOR PROSPECTIVE USERS

The interactive graphics system described in this article has been implemented on a CDC 3300 computer in conjunction with the Tektronix T-4002 Computer Graphics Terminal equipped with a joystick graphic input device. The CDC 3300 at OSU operates in a time-sharing mode utilizing a satellite PDP-8 as an I/O buffer. The programming language used is mainly FORTRAN

IV; however several utility routines, string processors, and the entire data filing and retrieval package are written in COMPASS. the assembly language for the CDC 3300. One could, of course, get along without the last mentioned data package, but it is surprising how often data are replotted, especially during the *preparation* of a demonstration. To *calculate* and plot a family of 10 curves for the equation

$$x(t) = A*\sin(\omega*t) + B*\cos(\omega*t)$$

at 100 points requires about 4.5 seconds of CPU time, while to *retrieve* and plot the same curves requires only 2.5 seconds. The more complicated the calculation, of course, the greater the benefit of retrieving the data.

The hardware requirements for development of an interactive graphics system of the type described here are (1) a time-sharing computer facility with sufficient memory capacity for about 10K instructions and the equivalent of 5K 24-bit words of storage, in addition to the requirements of the data generator, (2) a memory scope interactive graphics terminal capable of displaying both graphic and alphanumeric information ('Interactive' here implies only that some mechanism for inputting standard alphanumeric information be incorporated), and (3) a communications system capable of 1200 or perhaps 600 baud transmission rate. The latter is not a strict requirement, but our experience indicates that 110 baud would be totally inadequate for classroom use. In addition, it appears that some mechanism for graphic input may be indispensable.

It is doubtful whether at present many of our programs would themselves be directly useful to another user. However, considerable effort has and is being directed to the preparation of careful documentation and flow charts for those components that seem to have reached a reasonably stable state. When these are available (est. date: Sept. 1, 1971) a qualified programmer and highly interested supervisor should be able to develop a system similar to ours in from 3 to 6 months if basic routines are already available to drive the particular terminal. Flow charts and documentation (when completed) and/or program listings will provide to anyone requesting them. Contact the author or Mrs. J. Baughman, Oregon State Computer Center.

REFERENCES

1. *Computers in Undergraduate Education: Mathematics, Physics, Statistics and Chemistry*, Science Teaching Center, University of Maryland, College Park, 1967, pp. 6, 17, 22-24, 39, 54.
2. R. P. Feynman, *Feynman Lectures on Physics*, Addison-Wesley, Cambridge, 1963, Chap. 9, Vol. I.
3. *The Computer in Physics Instruction*, Commission on College Physics, College Park, 1967, p. 51.

Chance and Thermal Equilibrium

JON M. OGBORN, F. ROBERT A. HOPGOOD,
and PAUL J. BLACK

INTRODUCTION

This paper describes a set of films made on the Stromberg DatagraphiX SD 4020 display cathode-ray-tube at the Atlas Computer Laboratory, using the GROATS package.[1,2] The work has been done on behalf of the Nuffield Foundation's Science Teaching Project which is currently developing a new syllabus for teaching Advanced Level Physics to 16-18 year olds in British schools. In the course, among other things, children are introduced to a simple random game with board and counters which illustrate how chance will arrange energy among the atoms of a simplified model of a crystal. The computer films are used to show results which are hinted at by the students' own practical exercises but which are unobtainable in the classroom time available. At the present time, the course syllabus is being taught to a small number of schools in a pilot study.

The films are used with a section of the course called "Change and Chance" which lasts for about 30 classroom periods. Its aim is to make intelligible to pupils the ideas behind the Second Law of Thermodynamics and to introduce the concept of entropy. The approach is through the statistics of molecular chaos. This has been chosen mainly because it is thought that this gives a clearer insight than other approaches but also because of the growing importance of statistical thinking in physics and other subjects. The main statistical concept introduced is that what happens in many ways will happen often. Instead of *calculating* what will *probably* happen, random games are used to *show* what will probably happen. It is not expected that the course will produce competent thermodynamicists. However it is hoped that pupils will be able to recognize the fundamental difference between reversible and irreversible processes. They should understand that temperature is the quantity that indicates when thermal equilibrium is reached and that heat flow is always from hot to cold. Finally they should realize that the one-way

Chelsea College, University of London, London, England. Atlas Computer Laboratory, Chilton, Didcot, Berkshire, England. Department of Physics, Birmingham University, Birmingham, England.

nature of heat flow can be accounted for by chance alone. It is simply what is very likely to happen.

The earlier periods in the Change and Chance section are intended to show the pupil what a one-way process is. How it is always a process involving heat exchange and is, in general, a "spreading" process. The social consequences of the Second Law are examined by looking at the consumption of fuel in the world. The idea of thermal equilibrium is introduced, and temperature is defined as a measure of thermal equilibrium.

At this point a digression is made to consider diffusion. The pupils are introduced to a simple random game of moving counters on a board. This is used to show how chance alone brings about diffusion of gases to a uniformly spread equilibrium. Molecules that can spread into a larger space will do so. This need not be because they are pushed into the empty space but can simply be the result of chance. There are more ways of arranging the molecules when they are spread, and only rarely will chance take them back to the original unspread condition. The larger the number of molecules, the less likely is it that chance will soon produce a spontaneous reversal of diffusion. This digression will give the pupils experience of thinking about chance and about the number of ways things can happen. It also introduces them to the idea of playing random games to find out what chance will do.

THE MODEL

The model chosen to represent heat exchanges between atoms and molecules is the "Einstein solid." This consists of a regular array of harmonic oscillators, sufficiently strongly coupled to exchange energy, but not so strongly coupled as to disturb the energy levels of each oscillator. The energy levels are equally spaced and all energy exchanges are performed in multiples of a basic quantum of energy E ($E = h\,\nu$, where ν is the oscillator frequency, h = Planck's constant). If the model is chosen to imitate a crystal with a regular array of oscillating atoms, there is the further advantage that the sites of the lattice can be numbered and counted, even though the atoms themselves are indistinguishable and cannot be labelled in principle. The quanta, of course, are not distinguishable. Boltzmann himself used a similar model, considering lumps of energy held by atoms, long before quantum ideas were discovered.

The students are then allowed to play a random game simulating the model on a board marked out into 6 x 6 = 36 squares, each square representing an "atom" in this hypothetical crystal. Pieces can be placed in the squares on the board to represent quanta of energy, each piece representing a quantum of energy. As each square must have an integral number of pieces, the game exhibits the properties of the Einstein solid.

A move in the game represents the exchange of one quantum of energy between two atoms. A move can be defined as follows:

(1) A square is chosen at random, and a piece is removed from that square.

(2) A second square is chosen at random, and the piece is placed on this square.

(3) If a move initially chooses a square having no pieces on it then the move is "illegal" and no action is taken.

A proof exists that these rules simulate the exchange of *indistinguishable* quanta even though the pieces are distinguishable in principle. Students may want to exchange energy only between adjacent atoms feeling that this is more realistic. This is discouraged as problems occur at board edges. It is simpler to suggest that the jumps of a quantum in the random game might be the result of several steps in a real crystal. The game is not intended to be realistic; it just behaves as a model of a crystal.

Students are given 36 pieces and are allowed to position them on the board in any order. Using a pair of dice to choose squares on the board randomly, a number of moves in the game are made. After about 20 moves, the students are asked to plot on a histogram the number of atoms with 0, 1, 2, 3 etc. quanta. If the board is initially set with one piece on each square then the histogram initially would be like Figure 1. After 20 moves it might be like

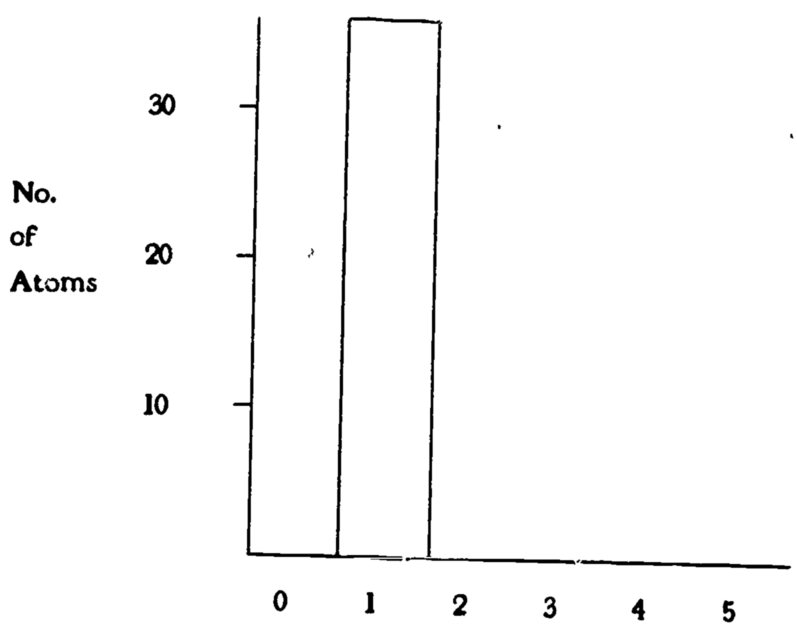


FIGURE 1 Initial distribution of quanta on 6 x 6 board.

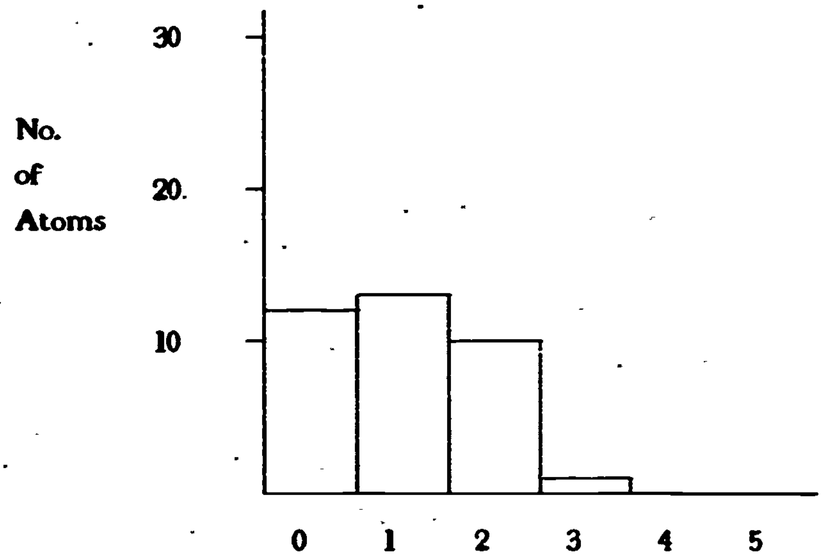


FIGURE 2 Possible distribution of quanta after 20 moves on 6 x 6 board.

Figure 2. No reconizable pattern will have appeared over the whole class, and, if starting positions are very different, results may vary widely. The game is continued for a further 100 moves and the histograms redrawn. Now many students will have histograms that have a shape similar to Figure 3. That is, the number of atoms with no quanta of energy is larger than the number with one quanta and so on. The results around the classroom will show substantial fluctuation, and it is worthwhile taking the average of the whole class. This should have a form similar to Figure 4. It now seems likely that the ideal or average distribution will consist of a histogram where the ratio between atoms with n quanta compared with $n + 1$ quanta is 2 for the case where the number of atoms and quanta of energy are equal [the ratio is in fact (N + q)/q for N atoms and q quanta].* It is at this stage in the teaching that the films made on the SD4020 are introduced.

**Editor's note* [RB]: This model corresponds to the classical harmonic oscillator with no zero-point energy, so that the oscillator energy in each category is

$$E_n = nh\nu$$

and the ratio of statistical weights between categories is given by the Boltzmann factor

$$N_n/N_{n+1} = e^{h\nu/kT} = R$$

The average energy is given by

$$E = \frac{h\nu}{R-1} = (q/N)h\nu$$

$$\therefore R = (N+q)/q$$

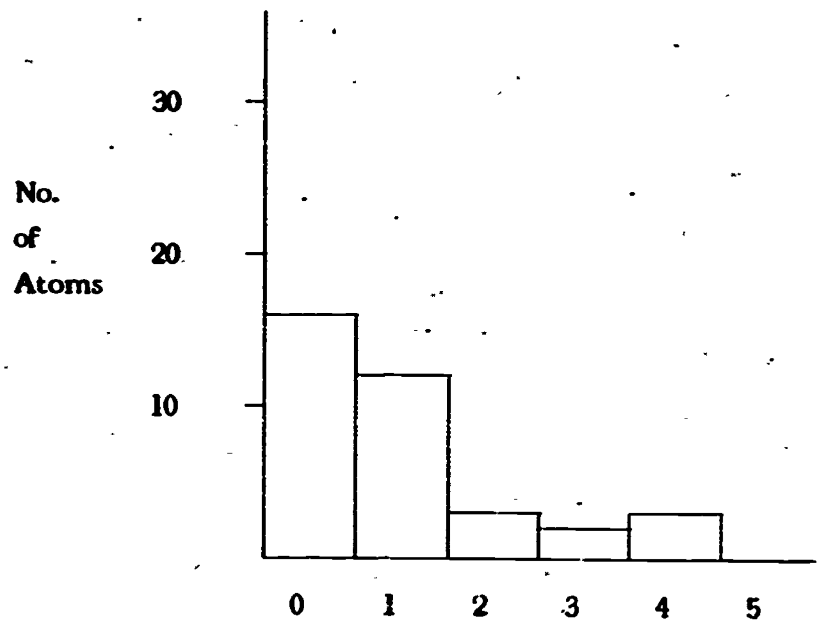


FIGURE 3 Possible distribution of quanta after 120 moves on 6 x 6 board.

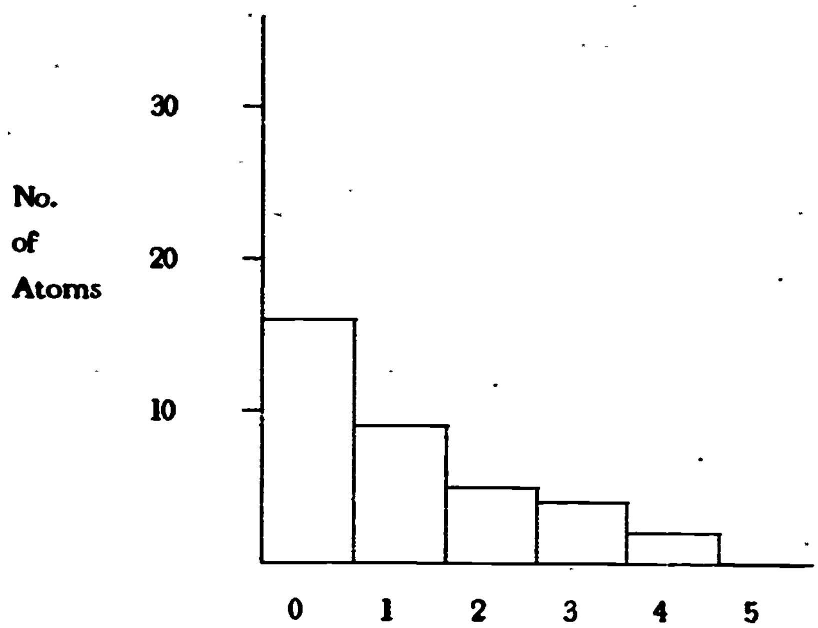


FIGURE 4 Average distrbution of quanta after 120 moves over a class of 30 students.

COMPUTER FILMS

Since the student may well be unsure that the game has progressed to a steady state and, indeed, may even be skeptical of the existence of a steady state, the computer films are introduced to resolve his doubts. They are as follows:

Film 1

The computer game uses a "board" 30 x 30. The starting point for the first game is with one quantum of energy on each atom. The film makes the initial moves very slowly so that the student can satisfy himself that the film is playing the same game. Later the speed of playing is accelerated, and a run of 20,000 moves is played. The initial and final frames of this sequence might be as in Figures 5 and 6. Taking the average of the last 10,000 moves, it can be seen that $N_n/N_{n+1} = 2$, where N_n is the number of atoms with n quanta of energy.

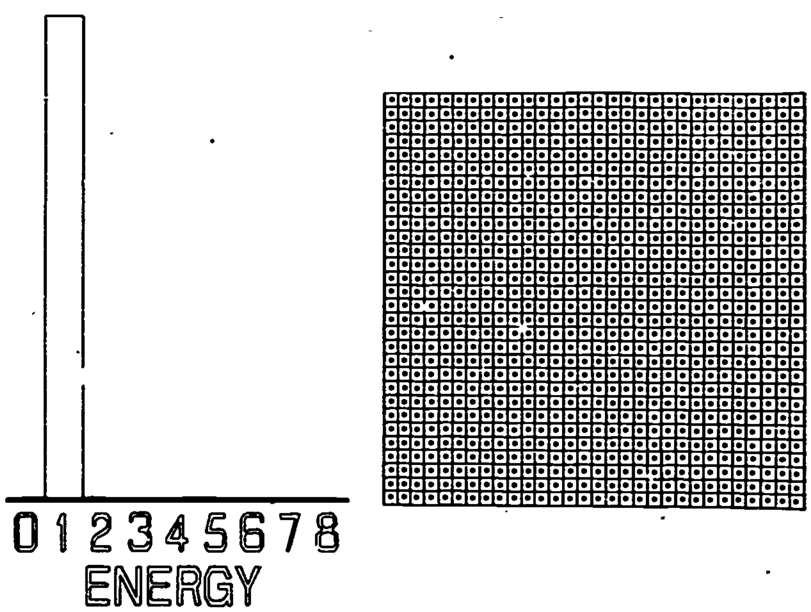


FIGURE 5 FILM 1 Initial board position and distribution 30 x 30 students.

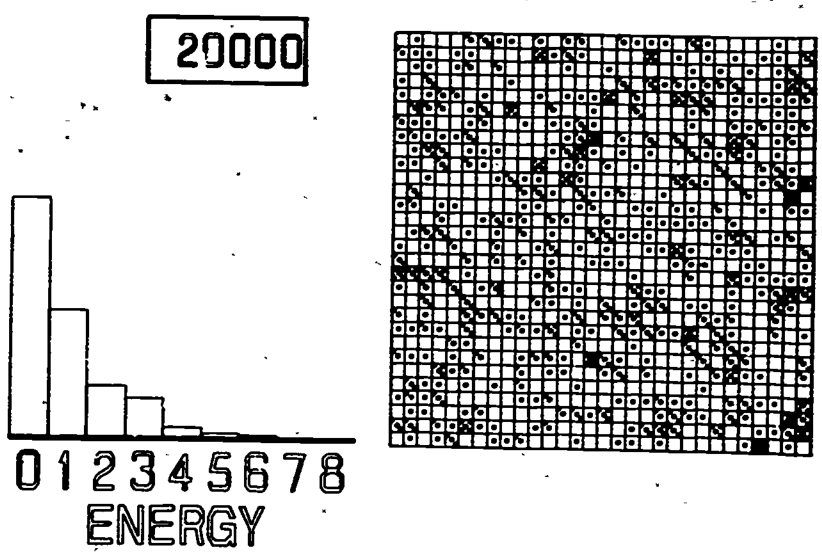


FIGURE 6 FILM 1 Final board position and distribution.

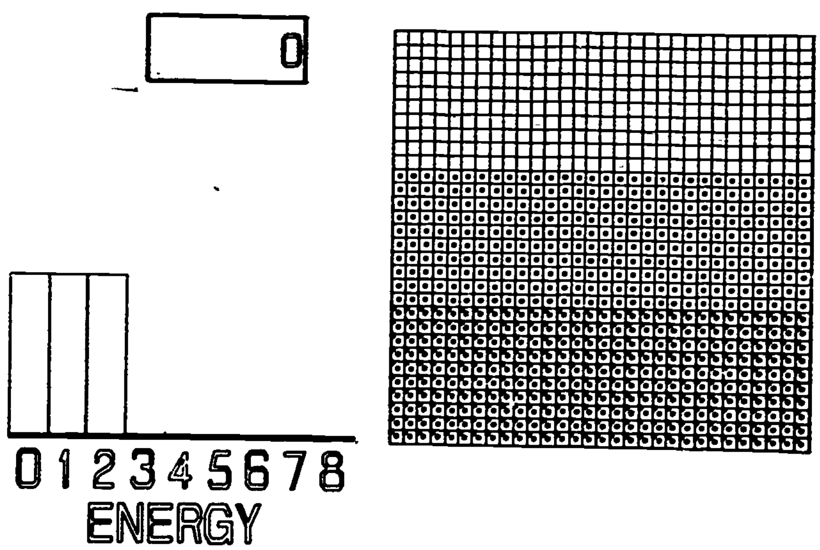


FIGURE 7 FILM 2 Initial board position.

Film 2

The game is repeated with 900 atoms and 900 quanta as before, but the initial board setup is changed to Figure 7. The game is replayed and the same ratio obtained.

Film 3

The game is repeated with 625 atoms and 625 quanta. Again the ratio $N_n/N_{n+1} = 2$ obtained.

These three films bring out the following points:

(1) A large group of atoms reaches a steady distribution after some time.

(2) The shape of the distribution does not depend on the starting position.

(3) The shape of the distribution is the same as long as the proportion of atoms to quanta remains the same.

(4) The shape of the distribution of number of atoms is a curve of constant ratio for equal steps of energy; that is, an exponential distribution.

So far, the results given above have been found as empirical facts of the computer analogue. The films are "thermodynamic experiments with a computer." Whether real systems behave the same is for further experiment to show. It is important at this point to ensure that the students notice the fluctuations of the distribution at equilibrium. The equilibrium is dynamic, and if the constraints are altered (see later) the fluctuations will "drive" the distribution to a new equilibrium.

Film 4

The game so far has suggested that *many* atoms have zero energy, fewer have one quantum, *fewer* still have two quanta. We would like to reword this as "that any one atom has no energy more often than it has one quantum" and so on. In putting the results this way, a jump has been made which is not obvious. This film therefore chooses a specific square and keeps a count of how many quanta it contains over a long period. The result of *Film 4* shows that if, on average, many atoms have energy E and half as many have energy $2E$, then any one atom will have energy E twice as often as it will have energy $2E$. A typical frame of this film is shown in Figure 8.

This is an important result and it is equally important to get the student to understand why this takes place. If we have an equal number of atoms and quanta, there is a definite number of ways in which the quanta can be

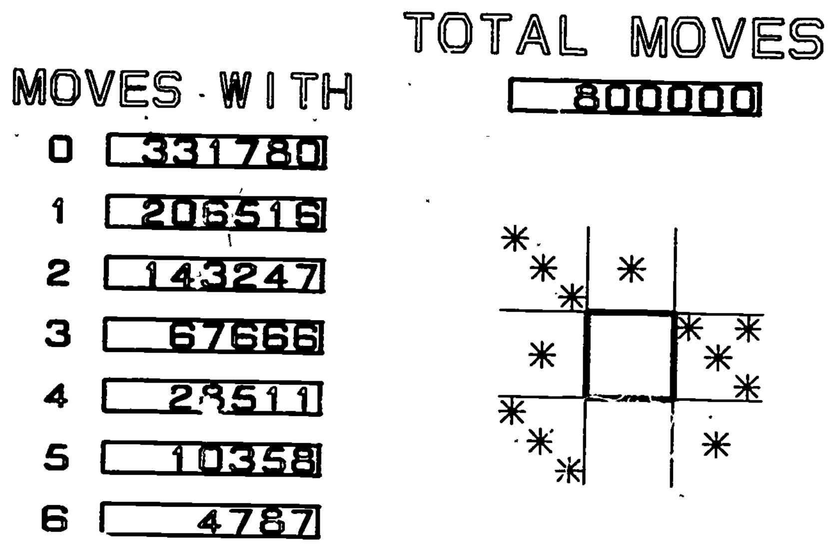


FIGURE 8 FILM 4 Frequency with which a particular square has a specific number of quanta.

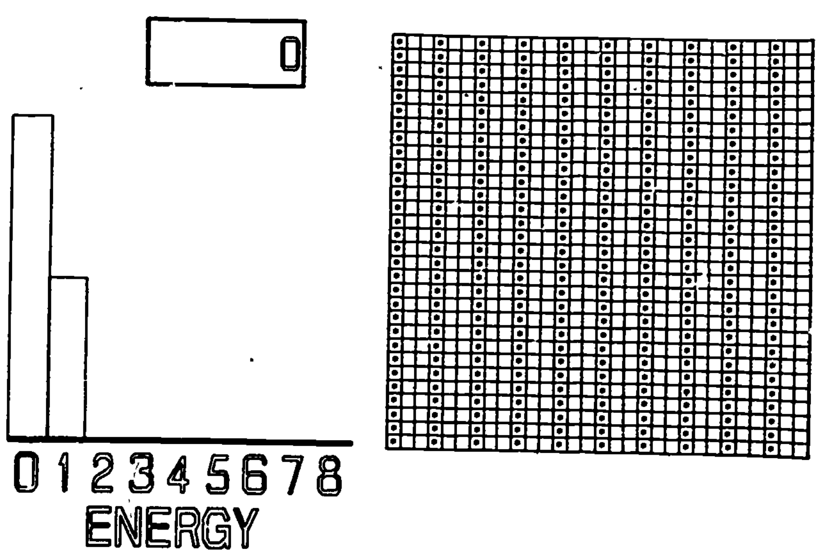


FIGURE 9 FILM 5 Initial cold board position; every third atom has a quantum of energy.

arranged. If one atom acquires an extra quantum then the number of ways the rest can arrange their quanta is reduced to half its previous value when one quantum is removed (assuming a large number of atoms). A group of atoms sharing quanta may pass one extra quantum over to one atom, or they may not. If they do, the quanta left can be arranged in half as many ways as if they do not. If moves are being made randomly then a move that removes a quantum from the specific atom will occur twice as frequently as before. Consequently a single atom will have energy nE twice as often as it has energy $(n+1)E$.

Film 5

It is now necessary to consider a colder model. Assuming the number of atoms is kept the same (900) then we consider a board where the number of quanta has been reduced from 900 to 300 (N atoms sharing $q = N/3$ quanta). *Film 5* starts with the board set up as Figure 9 and runs the game until an equilibrium is reached. The ratio between the number of atoms with n quanta and $(n+1)$ quanta is now $R = 4$ (Figure 10). This ratio can be taken as a measure of temperature. The larger the ratio, the colder is the object.

Finally we wish to see what happens to our model when the hot crystal described in the first three films is allowed to interact with the cold crystal described in *Film 5*.

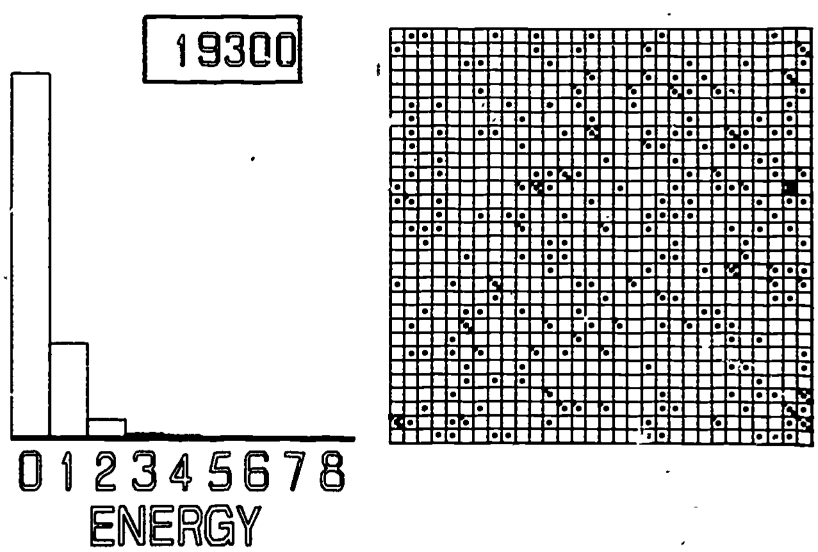
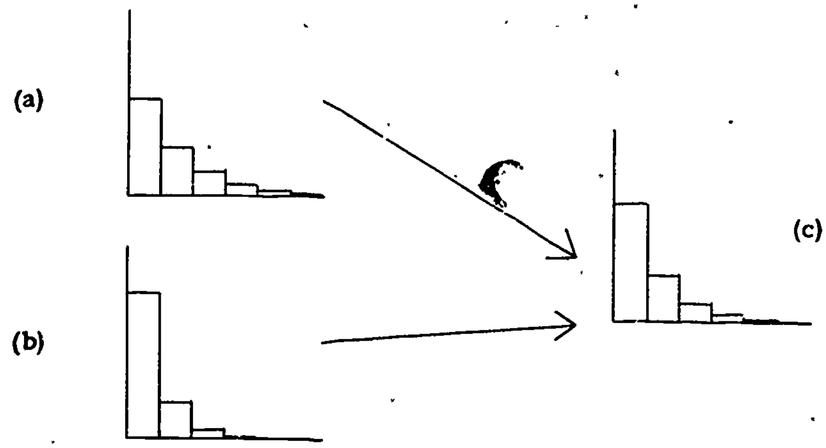


FIGURE 10 FILM 5 Equilibrium position for cold board.

Film 6

Initially the two boards with 900 and 300 quanta are allowed to achieve an equilibrium. The two boards are then brought together and moves may now take place between atoms on either board. As the film runs, quanta gradually move from the hot board to the cold board until an equal number of quanta appear on each board. The histograms for each board move from their initial distributions to the same distribution having a ratio R = 2.5 between columns (Figure 11).



(a) N atoms N quanta R=2

(b) N atoms N/3 quanta R=4

(c) 2N atoms 4N/3 quanta R=2.5

FIGURE 11 FILM 6 Distribution change when hot and cold board are merged.

The exchange of energy between these boards is a model of heat flow. The temperatures of the two boards have become equal. Heat flow from hot to cold has been demonstrated. This may look like an obvious result which does not require this amount of effort. However the important point is that this flow came about by random shuffling. Heat flow occurred spontaneously by chance. If the cold board receives an extra quantum then it now has four times as many ways of arranging quanta. Whereas the hot board only has its number of ways of arranging quanta reduced by a factor of 2. The "number of ways" of arranging quanta in a system is known as the *thermodynamic probability* of the system. If, initially, the thermodynamic probabilities of the hot and cold systems are separately W_H and W_C, then the combined boards have a thermodynamic probability of $W_H W_C$ initially. After the movement of one quantum from hot to cold, this becomes $(\frac{1}{2}W_H)(4W_C) = 2W_H W_C$, and there are twice as many ways of distributing quanta after the flow as before. And since this configuration can occur in twice as many ways it will occur twice as often.

Flow will continue from the hot board to the cold board until the thermodynamic probability of the combined system reaches a maximum. Additional flow of energy from one board to the other in either direction results in a less likely state of the combined system, i.e., one with smaller thermodynamic probability. Hence, it is less likely to occur. As mentioned before, fluctuations can be noted about the condition of maximum probability, i.e., thermodynamic equilibrium, but for samples this large they are relatively small.

The final part of the Change and Chance section returns to the real world and introduces the Boltzmann constant and entropy. The model, it is hoped, will have made this piece of work much easier. The statement that entropy has a powerful tendency to increase is seen to be equivalent to saying that a system spends most time in conditions that can arise in many ways, i.e., in the state of maximum thermodynamic probability.

Further details of the project can be obtained from J. M. Ogborn, The Nuffield Foundation Science Teaching Project, Chelsea College, 88/90 Lillie Road, Fulham, London SW6, England. The films will be distributed commercially; further information can be obtained from Michael Butler, Penguin Education, Penguin Books Ltd., Harmondsworth, Middlesex, England.

REFERENCES

1. F. R. A. Hopgood, *ALGOL GROATS Manual*, Atlas Computer Laboratory, Chilton, Didcot, Berks., U.K., 1969.
2. F. R. A. Hopgood, "GROATS, A Graphic Output System," in *Proceedings of the Eighth Annual UAIDE Meeting, 1969.* Users of Automatic Information Display Equipment, Coronado, California, 1970, pp. 401–410.

III

SIMULATION MODE

Introduction

The scientist's credo is triune; it rests upon Induction, Deduction and Proof. The first and last of this trinity must, in the natural sciences, entail experimentation. Hence, the importance of the simulation mode, in which the computer becomes a device to simulate physical reality in the form of governing equations, laboratory experiments or probabilistic processes.

In the first paper Lindsay literally uses the computer to simulate the operation of a multimeter as the student makes "measurements" of electrical quantities–current, potential, resistance–in an idealized dc circuit portrayed on his CRT. Ehrlich's work somewhat overlaps the computational mode and also presents an interesting method for generating diffraction patterns and wavefunctions of a hydrogen atom by means of a random process. The experimental aspect of simulation is again stressed by Katzper in his discussion of ray tracing to simulate optics experiments, whereas Wiley attempts to simulate the thermodynamic behavior of Ising lattices by Monte Carlo (random number) methods. Despite this persistent dichotomy of determinate versus random processes that we find in simulation, these four papers preserve its characteristically experimental approach.

A great advantage of simulation is that one can "simulate" quantities which do not, in the customary sense of the word, exist, such as electric or magnetic field lines, streamlines and "streak" lines. To carry this still further, while the simulation of a mechanics laboratory might allow the student to work in a universe where Newton's laws of motion hold, it might also allow him to specify non-Newtonian laws. Or the simulation can present the student with possibilities of experience such as relativistic phenomena which are not available normally, even though the laws are those of the "real" world.

Both Taylor and Stoner present papers that reflect this approach. Taylor describes a simulation of the space-time viewpoint of relativity in which the student is given a space-time world (coordinate system) on the face of a CRT with which he can interact dynamically, inputting events and worldlines and calling on Lorentz transforms. Stoner's work is a detailed examination, via desk-top computer, of the transformations that lead to the striking visual effects in a film on relativity done by Taylor himself and Judah Schwartz of the MIT Education Research Center. Yet, despite this connection, the two papers are quite different in their thrust and presentation and, as such, are complementary. This should prove interesting, and the reader may find himself unable to agree with Taylor's choice of a title.

CAI Physics Experiments

ROBERT E. LINDSAY

INTRODUCTION

This paper describes an exploratory investigation of a novel instructional method for physics and similar fields of study. Basic to this method is the exploitation of the *conversational* capability of a computer, through which the student is enabled to attack problems using a direct experimental approach—something not possible with any kind of conventionally constructed materials.

In our work the conversational capability is provided by an IBM 1500 Instructional System equipped with cathode-ray-tube terminals, light pen, and keyboard input. The 1500 system is specially programmed to operate in an interactive or conversational mode. The computer programs, written in the COURSEWRITER II language, provide for response to a wide variety of student input.

In the following we will give a general discussion of the instructional exercises with regard to their content, mediation (mechanics of student-computer communication), scoring, and control. We will also present a detailed description of several of the exercises along with some results of our evaluation of the exercises through student use.

DISCUSSION

The exercises are at the level of introductory college physics. They have been designed as a supplement to a conventional college physics course based on a typical introductory text in classical electricity and magnetism. They comprise the following twenty simulated experiments covering major concepts in a semester-length course:

1. *Cavendish Torsion Pendulum*—the moment of inertia is to be determined from observations of the period of free oscillation and the angular displacement in the presence of a static charge.

IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

2. *Coulomb Scattering*—the number of atoms per unit area of a metal foil is to be determined from measurement of particle deflection.

3. *Electrostatic Deflection*—the mean particle velocity is to be determined from measurement of particle deflection.

4. *Millikan Oil Drop*—the number of charges on the drop is to be determined through measurement of the velocity of the drop in various retarding fields.

5. *Gauss' Law*—the electric field at the center of a plane grid is to be made null by appropriate positioning of a point charge and two conducting spheres.

6. *Dielectric Constant*—the relative dielectric constant of a slab, which is subject to applied electrical and mechanical forces, is to be determined by observation of its motion between parallel conducting plates.

7. *Capacitance*—the dimensions of a capacitor with one movable plate are to be determined through measurement of its displacement under an applied force.

8. *Power*—the energy absorbed per cycle by a diode is to be determined from measurements of voltage and current.

9. *RC Circuit*—the values of R and C are to be determined from measurement of transient current.

10. *Multiloop dc Network*—the values of resistance are to be determined from measurements of voltage and current.

11. *Galvanometer*—the emf and internal resistance of a battery are to be determined from measurements with a tapped resistor and galvanometer.

12. *Thomson e/m*—the mass of an atomic particle is to be determined from measurement of its deflection in adjustable electric and magnetic fields.

13. *Oscillation of a Magnetic Dipole*—the magnetic moment of a dipole is to be determined from observation of its natural frequency in a magnetic field.

14. *Force on Current-Carrying Conductors*—the spring constant for a simple meter configuration is to be determined from observations of current and needle deflection.

15. *Betatron*—the number of particle orbits in a betatron is to be determined from measurements of Coulomb scattering.

16. *Ballistic Galvanometer*—the value of an unknown capacitance is to be determined from measurement of galvanometer deflection angle.

17. *Inductance*—the value of an inductance is to be determined from measurements of voltage and current.

18. *RL Circuit*—the values of R and L are to be determined from measurement of transient current.

19. *ac Network*—the value of an unknown reactance in a simple series circuit is to be determined from measurement of rms voltage.

20. *Resonance*—the parameters of a series *RLC* circuit are to be determined from measurements of power and frequency.

The content of our exercises is deliberately conventional so that we can more readily evaluate the novel features achieved by use of the computer as an interactive communications medium. The mechanics and structure of the communication process is sufficiently unconventional that we will describe it in detail. The student works at a computer-controlled terminal. The exercises assume that no physicist will be available to teach the student; the computer program permits the student himself, through conversational interaction with the computer, to deal with any questions that arise.

Some general communication problems must be solved to achieve effective conversational interaction of the student with the computer. When the student works on an exercise, the terminal screen presentation consists of diagrams of apparatus, words, and numbers. To perform an experiment he has to be able to connect meters in circuits, open and close switches, set voltages and currents, measure time and distance, etc., take data, and give answers. But how does the student know or find out where to touch or what to type to accomplish these things with the light pen or keyboard? How does he know what he must do and what he can ask for? Or, if he needs help, how does he get it? Also, how are the rules of procedure that are special to the problem in hand presented so the student is neither smothered by lengthy text nor left unenlightened or confused by too terse a set of instructions?

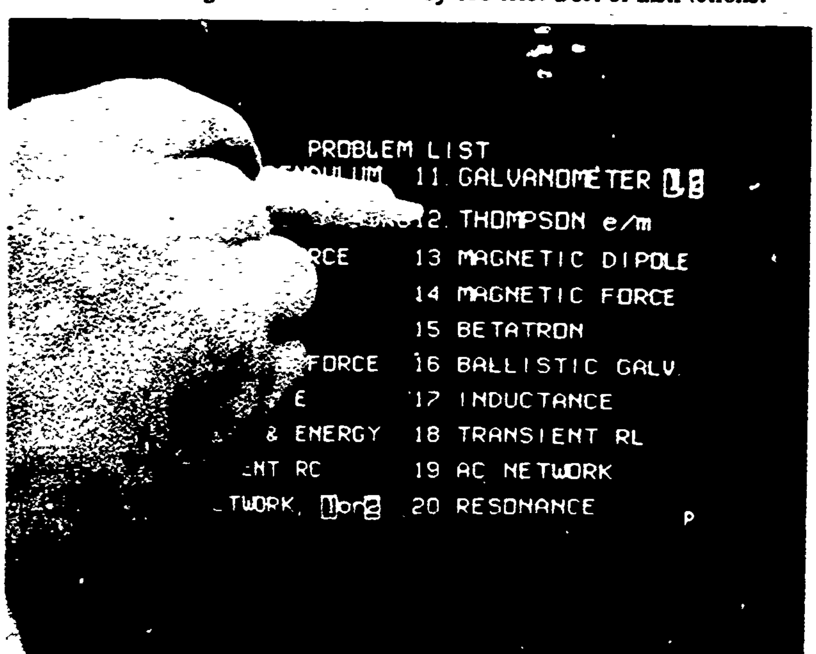


FIGURE 1 Initial screen presentation and exercise selection by light pen.

To answer these questions we will show what the student does when he works on an exercise. Figure 1 is the screen presentation after the student has signed on at the computer-controlled terminal. The "P" in the lower right-hand corner of the screen indicates the computer is waiting for a light pen response. By touching the numeral "12" the student has indicated he wishes to perform the Thomson *e/m* experiment. In Figure 2 the computer has responded with a display of the procedural instructions for the exercise. The words in capital letters, termed "control words," will be discussed in detail in what follows.

When the word "CONTINUE" has been touched with the light pen, the result is the display shown in Figure 3, the diagram of the apparatus. Since

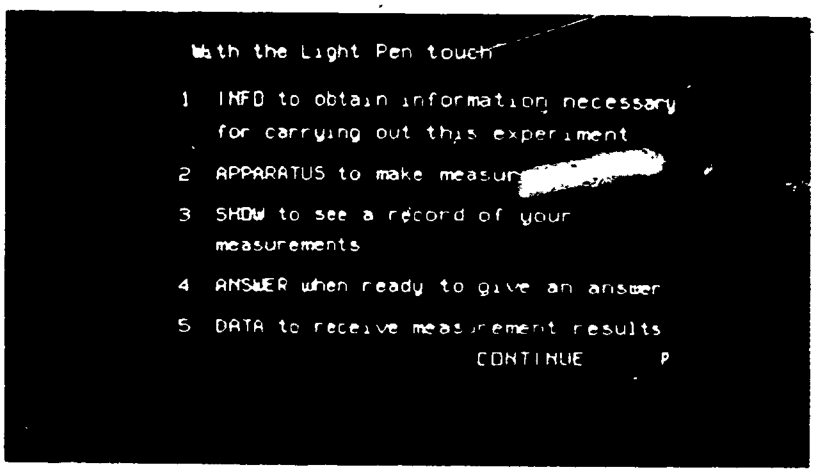


FIGURE 2 Procedural instructions.

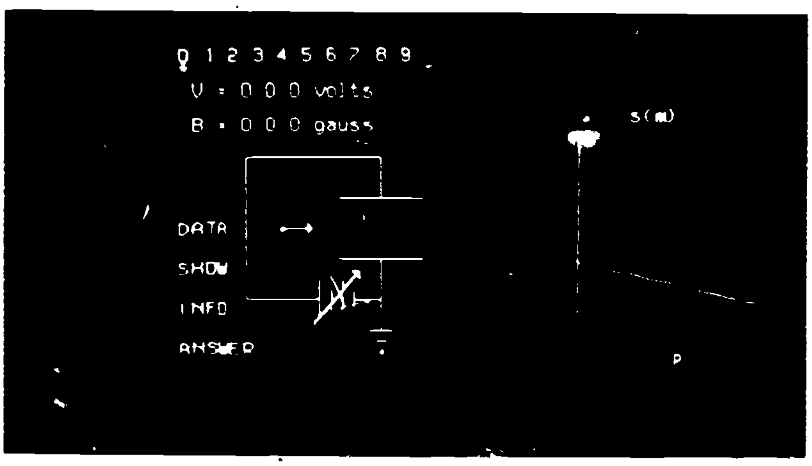


FIGURE 3 Display for the Thomson *e/m* exercise.

the student has previously studied a syllabus sheet (see Appendix), such things as dimension lettering and diagram labels have been omitted from the display for clarity. The display shows the deflection plate circuit, electron gun (arrow), and the screen of an evacuated tube. V and B indicate that electric and magnetic fields are to be varied to produce deflections of the beam of particles from the gun. In order to perform this experiment successfully the student must obtain zero deflection with both electric and magnetic fields present.

The student touches the control word "INFO" to learn how to proceed with the experiment; the resulting computer response is given in Figure 4, the list of available information. Figure 5 shows the result of touching the second

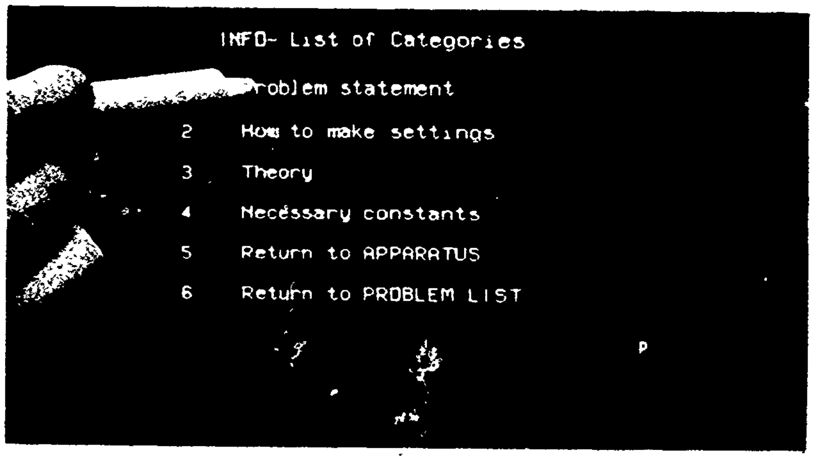


FIGURE 4 Response to "INFO" request, Figure 3.

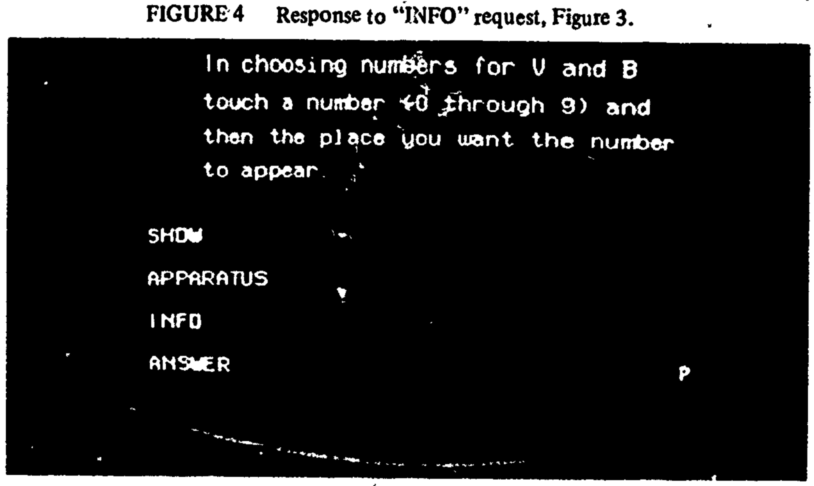


FIGURE 5 Response to "Problem Statement" request. Figure 4.

category in Figure 4. Some useful equations and indication of the notation to be used throughout the experiment are displayed when the category "Theory" is touched (Figure 4). By touching the fourth category, "Necessary constants," the student receives the experimental and physical constants associated with the apparatus.

The student now returns to the apparatus diagram, by touching the numeral "5" shown in Figure 4, and sets voltage to 50 volts according to the rule given in Figure 5. He then touches "DATA" (Figure 6); what he then sees is a beam

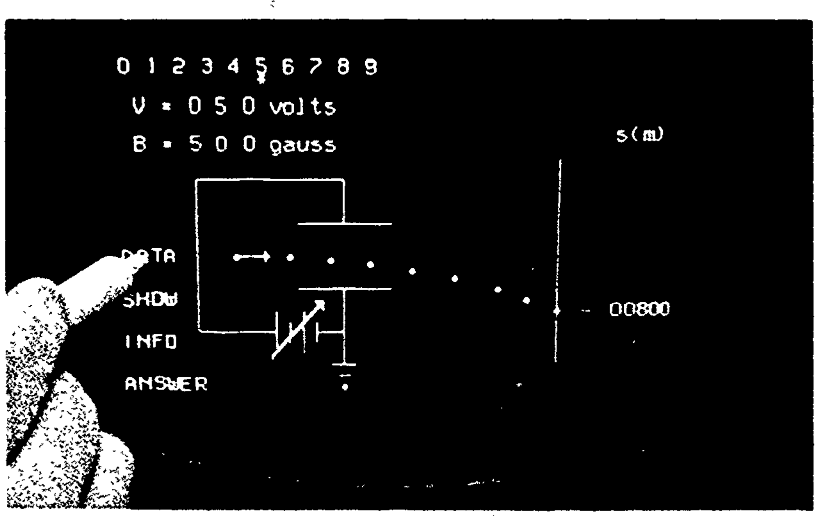


FIGURE 6 The Thomson *e/m* experiment.

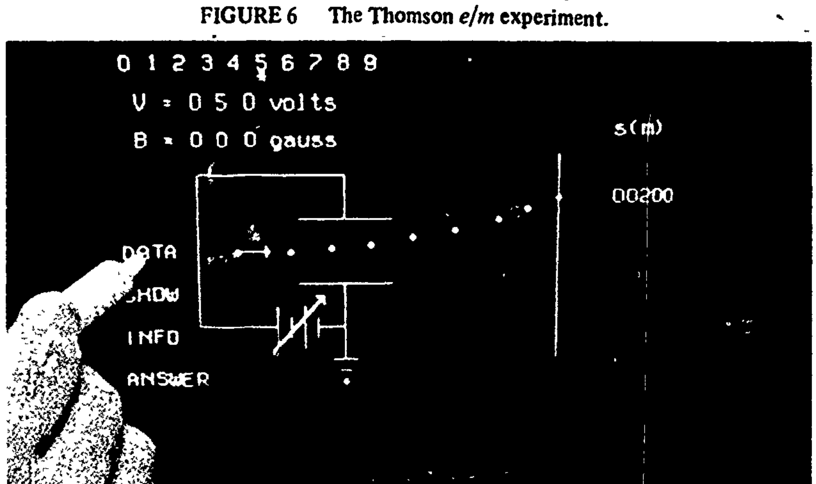
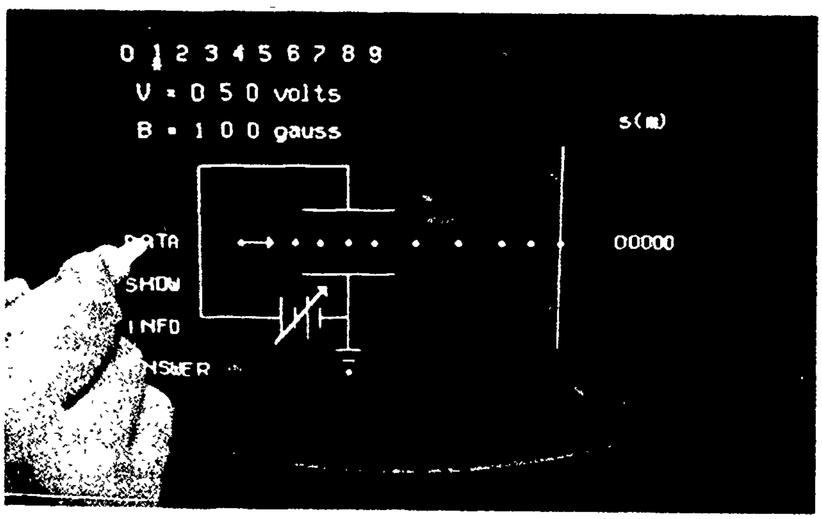


FIGURE 7 Effect of changing the magnetic field from zero (Figure 6) to (a) 500 gauss, and (b) 100 gauss.

of particles moving through the plate region, being deflected upward, and impinging on the screen 0.002 meter vertically from the center.

The magnetic field is now to be adjusted to achieve zero deflection. Figure 7(a) indicates the effect of setting magnetic field at 500 gauss with an electric field of 5000 volts per meter. By reducing the magnetic field to 100 gauss, the student obtains balance–Figure 7(b).

The student may touch "SHOW" to review the results of his measurements, and when he has computed his answer he can touch "ANSWER," the computer responding as in Figure 8. The small rectangle is a "cursor" which indicates where his typewriter input will appear. The "K" in the lower right-hand corner means that the computer is waiting for a typed (keyboard)



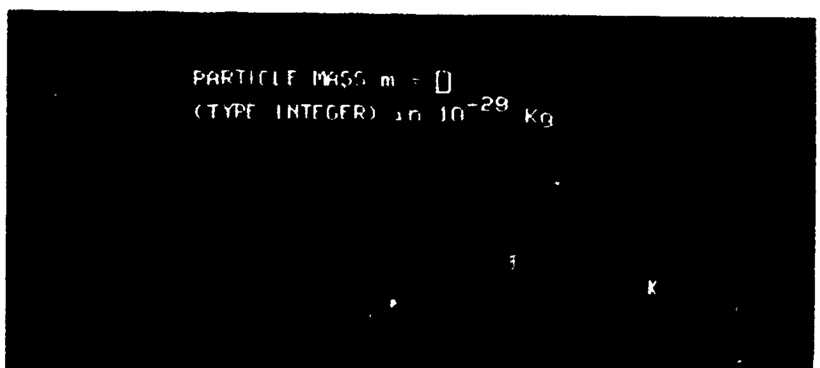


FIGURE 8 Response to student request for "ANSWER."

response. After the student types a value 2000, the computer evaluates his response as correct, answering "OK!"; if incorrect, the response is "NO!" and appropriate control words. After completing the experiment, the student may then return to the list of exercises (Figure 1).

In addition to the control words "INFO," "SHOW," and "ANSWER," the word "DATA" provides meter readings and experimental numbers, and the word "APPARATUS" displays the apparatus on which the student works. This set of five control words are all the instructions he needs to know. Some always appear on the screen; other, only when necessary. As he uses them, the student is instructed in the rules of procedure for the exercise. Since the student communicates with the computer in the same way whether he is doing a torsion pendulum experiment or a resonance experiment, he will not be burdened with learning special procedures for each exercise; and his mind will be on the physics of the problem rather than on the mechanics of communication.

There are also communication problems peculiar to the individual exercise. How is the experiment performed? How is it made clear to the student what he is doing? As seen earlier, the student works on an apparatus diagram which appears on the screen when the control word "APPARATUS" is touched. This diagram *changes* as he performs his experiment, connecting meters, opening and closing switches, setting voltages and currents, etc. There is a consistent set of conventions for communication with the diagram: meter connections, switch settings. and entry of numbers for voltages, currents, etc., are accomplished with the light pen, though numerical information *may* be entered through the keyboard.

To aid the student in remembering what he has touched with the light pen, the diagram usually changes when the light pen is pressed to some significant point on it; e.g., if a number is touched, an asterisk appears below the number; if a node in a circuit is touched, an asterisk appears next to it—unless a meter terminal "P" or "N" has been previously touched, in which case a "P" or "N" will appear next to the node indicating connection to the meter. In addition to helping him remember what he has done the asterisk or "P" or "N" on the diagram provides an unambiguous indication to the student that his command has been recognized and executed by the computer.

Thus far content and mediation have been discussed. At this point a brief look at scoring and control is in order. How is it possible to have a number of students take the same exercise without passing on the correct answers? On the other hand, how can they be encouraged to help one another? We deal with these problems by using the computer's random number generator, which randomly generates new problem parameters (such as mass in the Thomson *e/m* experiment) *each time* the exercise is taken so students can take the same exercise at the same time (each at a different terminal) with each student having a different set of answers. Furthermore, a student can be required to repeat a particular exercise, each time with a different set of parameters, until a specified performance level is achieved.

EXERCISES

Two exercises will now be described to illustrate further the points brought out in the preceding discussion. They have been chosen because, taken together, they demonstrate most of the experimental manipulations encountered in the rest of the exercises. These examples demonstrate the treatment of electric circuits.

The circuit diagram for the Multiloop dc Network (see Appendix) exercise appears in Figure 9. This simulates a meter, which may be connected as either a voltmeter or ammeter, and a two-loop dc circuit with an unknown voltage

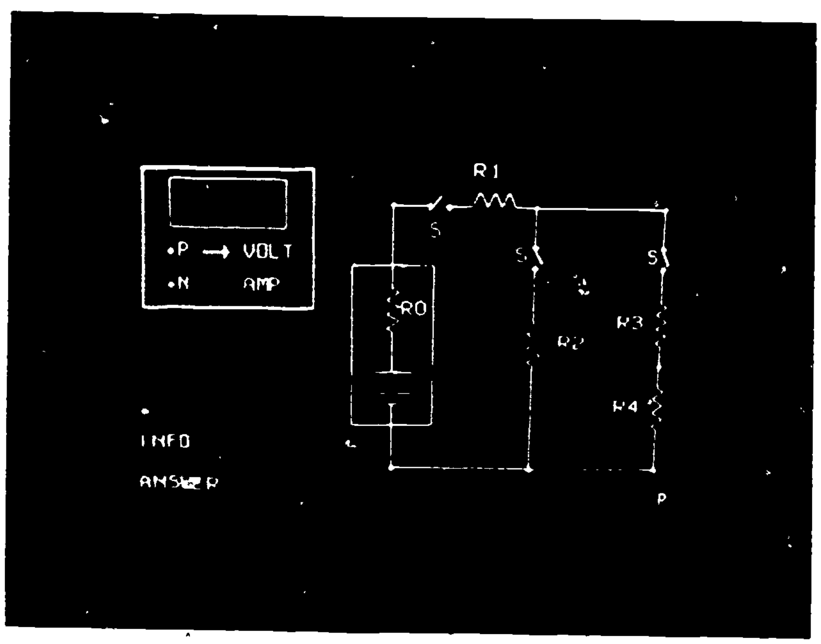


FIGURE 9 Apparatus for Exercise 10, Multiloop dc Network.

source and five unknown resistors. Three switches have been placed in the circuit for measurement flexibility. The student must determine through voltage and current measurements the resistance in ohms of each resistor.

The student first calls "INFO," which tells him to touch "VOLT" or "AMP" to set the meter, connecting its terminals to the circuit by touching "P" or "N" and then the appropriate node; to change a switch "S" he simply touches it. In Figure 10 the appropriate switches have been closed, and the meter is set as a voltmeter (arrow). Meter terminal "P" is now touched; and then the upper left corner of the circuit is touched, the "P's" and associated arrows signifying that the commands have been executed and the connection made. The negative terminal "N" of the meter (with an arrow) has likewise been connected to the lower left corner of the circuit, and the meter reads 7 volts. To obtain the loop current the meter, set as an ammeter (arrow), is connected in series as shown in Figure 11. The reading is 1 milliamp. Each

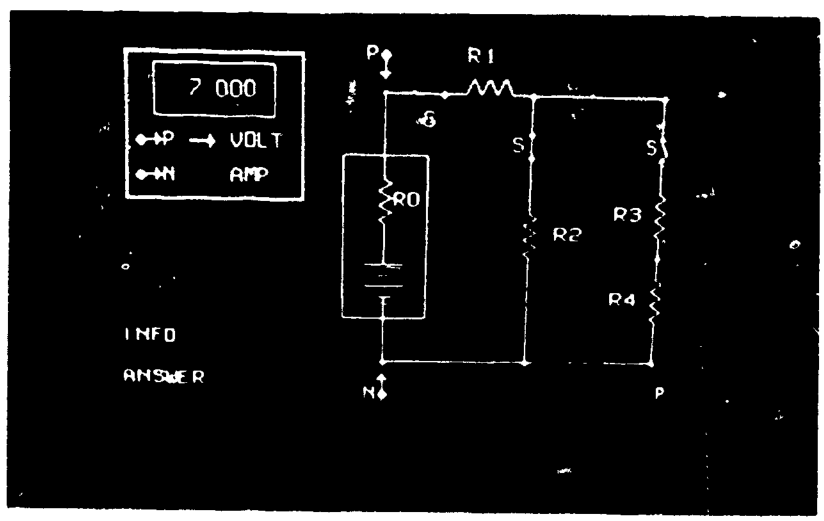


FIGURE 10 Multimeter of Figure 9 connected as a voltmeter.

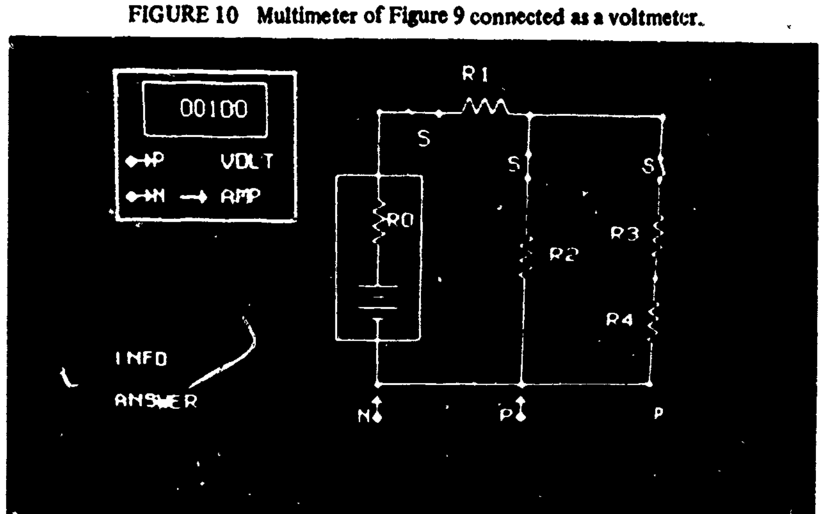


FIGURE 11 Multimeter of Figure 9 connected as an ammeter.

time the exercise is taken, the resistors are given different values by the random number generator.

When the student has made enough measurements to determine the resistances, he touches "ANSWER" and is instructed to "Touch Resistor." Figure 12 shows the effect of touching "R2." If the response is correct, the computer answers "OK" and replaces "R2" by its actual numerical value. The same procedure can be followed with the other resistors.

Figure 13 shows the simulated *RLC* series circuit and wattmeter for the

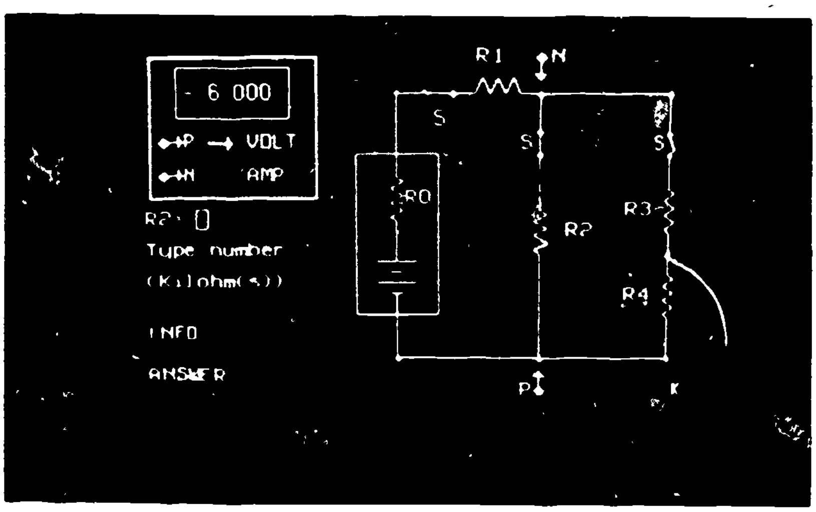


FIGURE 12 Response to student request for "ANSWER."

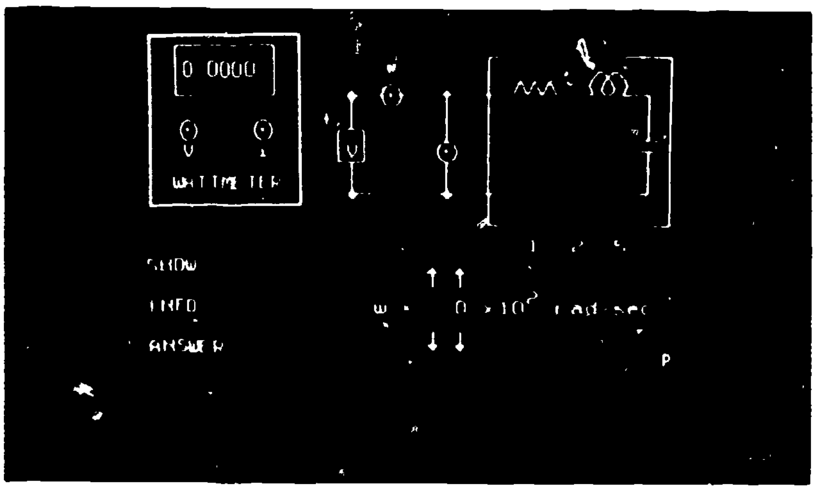


FIGURE 13 Apparatus for Exercise 20, Resonance.

Resonance exercise (see Appendix). In this experiment the student must vary the frequency ω of the source, take wattmeter readings, and determine the values of *R*, *L*, and *C*. The student can learn how to connect the wattmeter and vary frequency in the INFO section which provides rules for frequency variation, input voltage, and some useful theory.

The student begins by connecting the wattmeter in the circuit. He then sets the frequency and obtains a reading on the wattmeter. After taking several readings, he reviews his data by touching "SHOW" which includes an option ["CONTINUE," Figure 14(a)] for graphing his results [Figure 14(b)]. The

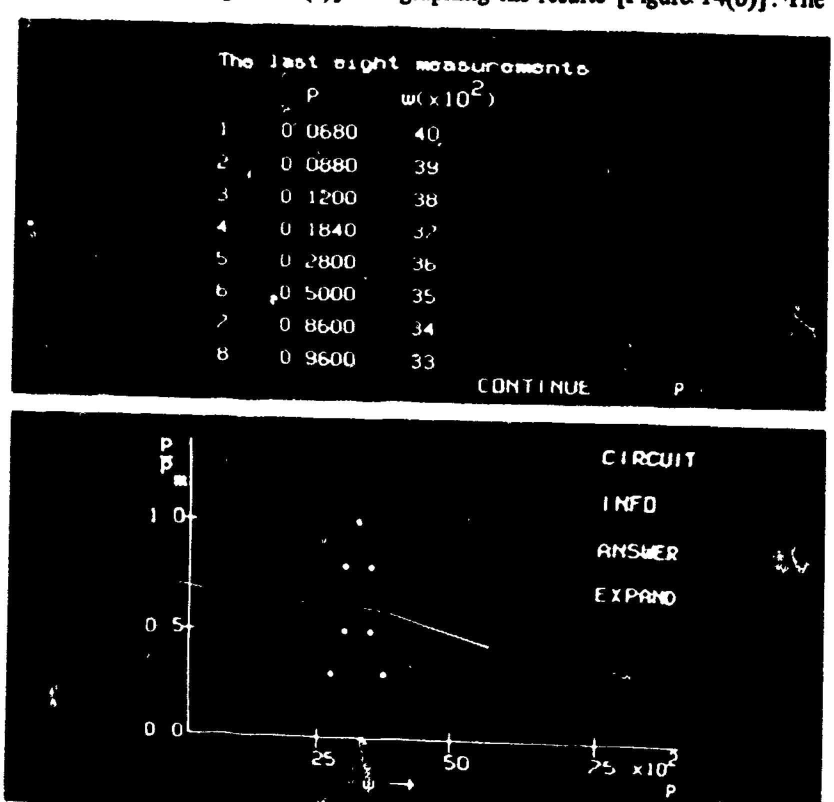


FIGURE 14. Response to student request for "SHOW," Figure 13: (a) data; (b) graph of data.

"EXPAND" option allows him to expand the scale of the graph about resonance for more detailed study, while "CIRCUIT," when touched, will return him to the apparatus diagram of Figure 13. As in the preceding example, the student can then indicate that he wants to give the answer for the value of one of the three circuit elements in Figure 13.

All the exercises have an experimental aspect. The student must understand the physical situation in order to know what data he needs and how to reduce these data. The materials are simulated experiments; although, strictly speaking, they are not simulations of the actual physical laboratory.

As he is being introduced to laboratory concepts, the student deals with a level of abstraction in that he communicates with diagrams of apparatus rather than the apparatus itself. Also, these simulated experiments go beyond introducing the student to the laboratory in that some of the exercises are of experiments that could not be performed in a real laboratory—because of equipment limitations, time, etc. Further, these physics materials are more challenging than ordinary homework; in a homework problem the data are given, and the student applies calculative techniques to obtain the answer. In the simulated experiments the student must obtain his own data as well as apply the proper formulas.

Now that the exercises have been described in some detail it will be of interest to look at some results from a preliminary study with students.

STUDENT EVALUATION

The physics materials to this time have not been used in conjunction with an actual university physics course, but an initial evaluation has been accomplished with volunteer students who have had at least an introductory college-level course in electricity and magnetism. Some results of this evaluation follow.

The students have had little difficulty with the mechanics of interaction with the computer. After becoming familiar with the terminal they have been able to use the control words to move within the particular simulated experiment and have performed the necessary manipulations to obtain data.

In many cases, however, the students have shown a lack of understanding as to what data are relevant to the successful completion of the experiment. They have not understood the physical situation. A typical student comment has been,

> "I had to understand the problem to get the right answer."

Given the necessary data the student can apply his calculative skills to obtain answers, but getting these data on his own seems to be a real obstacle.

A number of students have suggested the exercises be used with a regular course. One interesting comment is the following:

> "I feel the experiments should be a supplement to a theory course and used in conjunction with electrical labs. Also I would like to see some mechanics experiments which are impossible to carry out in a real laboratory (e.g., Bohr Theory)."

The student has recognized one of the reasons for developing these simulated experiments.

That the exercises have served as a review and fostered a better understanding of underlying physical concepts is seen in comments such as the following:

> "They helped me to remember concepts I had forgotten. I think I will remember them more easily now."
> "I was familiar (general knowledge) with all the experiments, but doing them clarified them and gave me an understanding of them."

Most of the students have indicated their experience at the terminal has been enjoyable. This is an important fact considering it is not uncommon today for a student's learning experience to be characterized by boredom and frustration. Below are some representative comments:

> "I found it quite interesting and fascinating."
> "I was very excited about using the machine in general."
> "I found the experience exhilarating."

In some instances the students have not been given adequate prior explanation as to what to do at the terminal, nor been asked to prepare for the exercise in advance. In these cases usually the experience at the terminal has not been so pleasant:

> "I spent most of my time trying to derive equations and not much time on the screen."
> "Not enough theory explanation."
> "It gave me a slight headache."

CONCLUSION

In this article a new instructional method for physics and related fields has been introduced. Physics has been chosen because it is typical of a broad class of studies—the physical sciences. What is learned through development and usage of these physics materials will be applicable in these other areas, too.

One purpose for the development of these exercises has been the study of the mechanics of student-computer interaction. How are materials developed such that it is obvious to the student in an instructional situation what he can do and what he can ask for? The results from our preliminary studies seem to indicate that these physics materials offer an approach to an answer to this question.

It is probable that the majority of those reading this paper are not familiar with Computer-Assisted Instruction. We hope that it has given some insight into the potentiality of this new medium for instruction.

Physical Simulations for an On-Line Computer-Controlled Oscilloscope

ROBERT EHRLICH

INTRODUCTION

This paper describes a set of programs that have been developed to generate animated pictures on a computer-controlled display oscilloscope. These simulations, intended for instructional use, constitute accurate representations of problems in mechanics, waves, thermodynamics, and quantum mechanics. The user at a teletype on-line to the computer, (a Digital Equipment PDP-6) types numerical values for a number of requested parameters. After this, the computer uses these numerical values to carry out whatever calculations are needed to generate the display.

Three kinds of use have been made of the simulations:

1. Preparation of slides and computer-animated films by directly photographing the oscilloscope screen.
2. On-line use as a visual aid by the lecturer. For this purpose, a teletype which is interfaced via a telephone line to the computer, has been installed in the lecture hall. A television camera, part of a closed-circuit TV system, is used to photograph the oscilloscope screen.
3. On-line use by the individual student, allowing him to perform, in effect, simulated experiments.

The programs have been written for a Digital Equipment PDP-6 computer having a Tektronix model 611 display oscilloscope. Pictures consisting of any number of points, line segments and characters can be displayed on the oscilloscope in one of two modes. In storage mode, a picture is continuously built up, and it remains on the screen until it is erased. To achieve animation, the computer generates new pictures (non-storage mode), or adds to an existing picture (storage mode), at a sufficiently high rate (thirty or more times per second).

Department of Physics, State University College, New Paltz, New York. Work supported in part by National Science Foundation, and performed while at Rutgers University, New Brunswick, New Jersey.

In addition to four topics in elementary mechanics (one-dimensional kinematics, projectile motion, collisions and rocket flight), visual displays have also been generated for a number of other topics.

SATELLITE ORBITS

In this simulation, the computer generates animated pictures of a satellite moving in the gravitational field of the earth or, alternatively, the earth-moon system. The satellite orbit is calculated iteratively, using a fourth-order Adams-Bashforth formula.[1] In this method, the computer calculates the position $\mathbf{r}_5$, and the velocity $\mathbf{v}_5$ at time t_5, in terms of the position $(\mathbf{r}_1 \ldots \mathbf{r}_4)$, velocities $(\mathbf{v}_1 \ldots \mathbf{v}_4)$, and the acceleration $(\mathbf{a}_1 \ldots \mathbf{a}_4)$ at four preceding equally spaced instants of time $(t_1 \ldots t_4)$, where $t_n = t_0 + n\Delta t$, and:

$$\mathbf{v}_5 = \mathbf{v}_4 + \Delta t/24\,(55\mathbf{a}_4 - 59\mathbf{a}_3 + 37\mathbf{a}_2 - 9\mathbf{a}_1) \qquad (1)$$

$$\mathbf{r}_5 = \mathbf{r}_4 + \Delta t/24\,(55\mathbf{v}_4 - 59\mathbf{v}_3 + 37\mathbf{v}_2 - 9\mathbf{v}_1) \qquad (2)$$

The computer then calculated the acceleration $\mathbf{a}_5$ using the inverse-square force law.

If the student types the appropriate numerical values for the initial position and velocity of the satellite, the computer generates displays of the corresponding orbits [see Figure 1(a)]. The student can also study force-distance relations other than the inverse-square law. To do this, the student types in the exponent in the force-distance relation. He can then observe a number of phenomena, such as:

1. Unstable spiral orbits resulting from exponents less than −2. The outwardly spiraling orbit, shown in Figure 1(b), is for an exponent of −3. (Lower initial velocities would cause the orbit to spiral inward.)
2. Precession of the perigee of an orbit, when the force-distance law is slightly different from inverse square, as shown in Figure 1(c), for an exponent of −2.2.
3. Unusual orbits resulting from large positive values of the exponent, as shown in Figure 1(d), for an exponent of +50.

The student can also observe orbits of satellites moving in the gravitational field of the earth-moon system. He can, for example, try to launch a satellite in a figure-eight orbit around the moon as in Figure 1(e). Another effect that he can study is the effect of lunar perturbations on a satellite launched in an initially circular earth orbit, Figure 1(f).

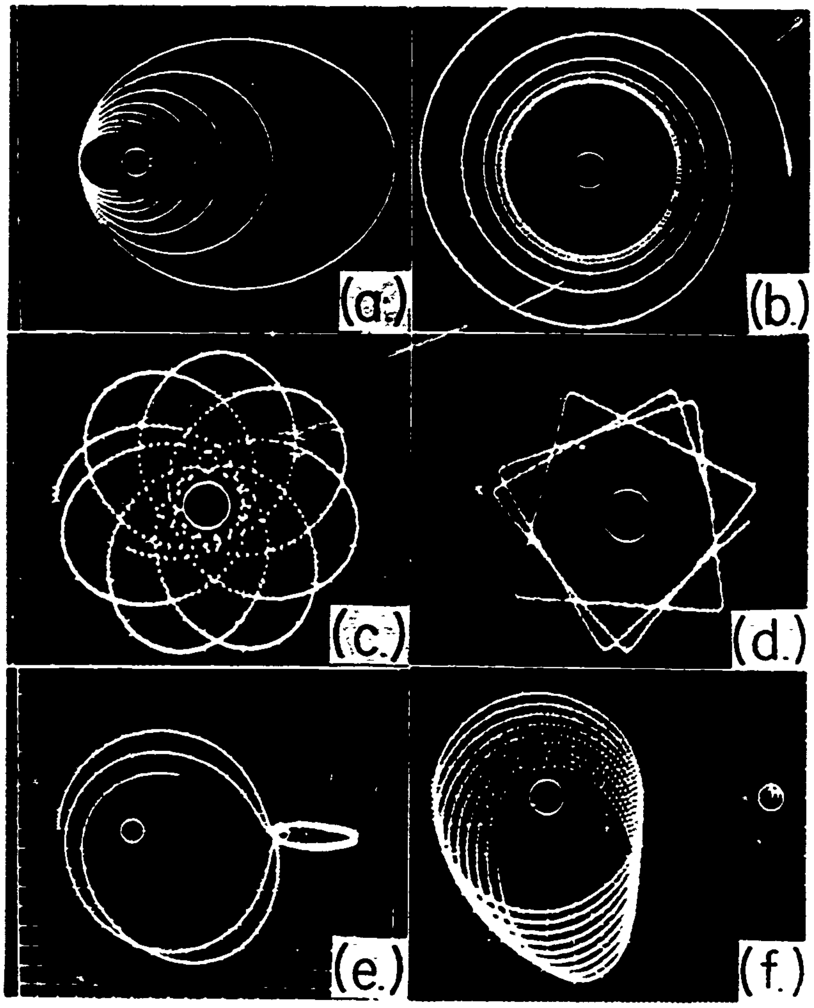


FIGURE 1 Satellite orbits. (a) Orbits having a constant increment in the initial velocity, (b) unstable orbit for a force law: $F \sim r^{-3}$, (c) precession of the perigee for a force law: $F \sim r^{-2.2}$, (d) orbit resulting from the force law: $F \sim r^{+50}$, (e) approprimate figure-eight lunar orbit, (f) lunar perturbations on a satellite in earth orbit. [In (e) and (f) the force law is inverse square and the earth-moon mass ratio is taken to be 20 rather than 81 to get a better-looking display.]

PERFECT AND IMPERFECT GASES

This simulation, involving the time evolution of a number of moving molecules confined to a box, allows the student to perform a variety of simulated experiments on perfect and imperfect gases. Each molecule is represented in an animated display by a moving point (the center of the molecule). The perfect gas consists of non-interacting spheres whose radius may be specified by the student. For the imperfect gas, a finite range attractive force is specified between each pair of molecules. Thus, for molecules of non-zero size, the student may select one of the two forms of the intermolecular potential shown in Figure 2. Other quantities that the student needs to specify include the volume of the box, the temperature of the gas, and the initial state of the gas. At the student's option, the initial velocities of the molecules are either all the same, or else they are chosen randomly. The same holds for the initial positions.

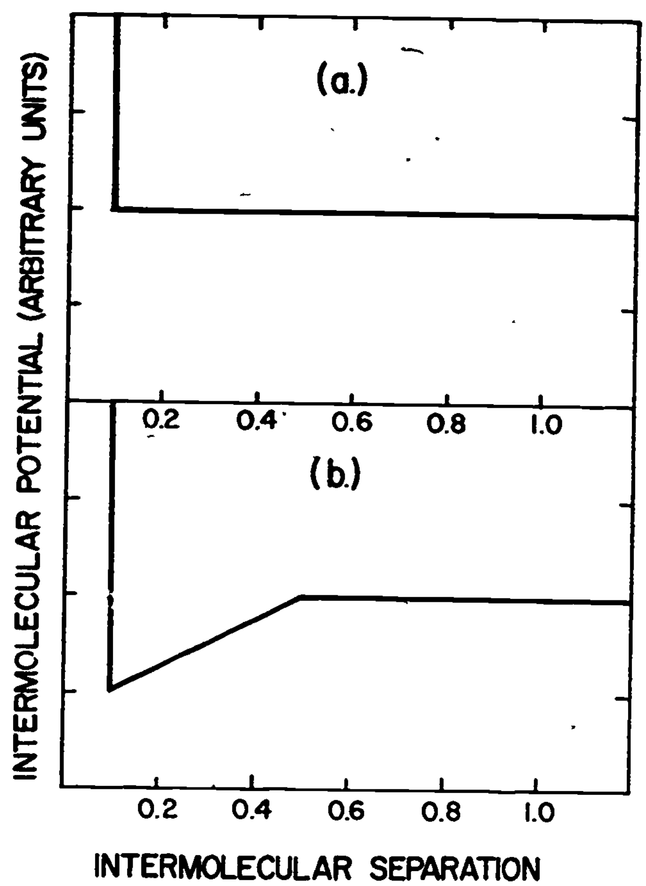


FIGURE 2 Assumed forms of the intermolecular potential between a pair of finite size gas molecules. (a) hard core repulsion only, (b) hard core repulsion plus finite range attraction. (The sides of the containing box have unit length.)

The computer generates initial velocities for the molecules, such that their average kinetic energy corresponds to a temperature specified by the student. As the collection of molecules evolves in time, the computer calculates the pressure exerted by the gas on the walls of the box from the forces that individual molecules exert during their collisions with the walls in some specified time interval. (The student specifies both the time interval and the number of molecules.) The pressure computed in the manner just described is subject to statistical fluctuations, whose magnitude depends on the number of molecules and the number of collisions made by each. The statistical fluctuations are fairly large unless the molecules are allowed to make a reasonably large number of collisions, since the number of molecules must be kept small (under ten), in order that the display remain reasonably flicker free.

The student can type in a range of values for the volume of the gas at a number of temperatures, allowing the computer to calculate the pressure in each case. In this way, the student can "experimentally" obtain isotherms for perfect and imperfect gases, such as those in Figure 3. The scatter of the "experimental" points is due to the statistical fluctuations noted previously.

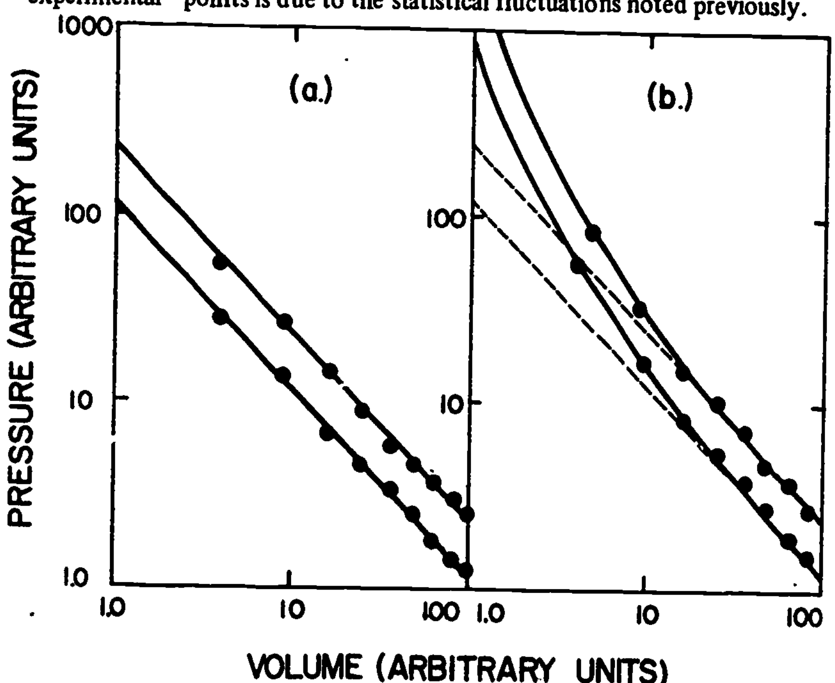


FIGURE 3 Isotherms obtained from computer-calculated pressures for a range of volumes at two different temperatures for (a) a perfect gas in which the molecules have zero size, (b) a perfect gas in which the molecules have finite size [potential shown in Figure 2 (*a*)].

The student can also study the relationship between pressure, volume, and temperature in other more qualitative ways. The student can specify the velocity of the walls of the box (normally taken to be zero). If the velocity is non-zero, the size of the box alternately expands and contracts. Alternately, the student can specify that the temperature of the gas be gradually raised (lowered) by specifying that either the intermolecular or molecule-wall collisions should be inelastic (explosive). If he specifies that the temperature of the gas is to be gradually lowered, the student can directly observe an animated display of a condensation process. In this case, an imperfect gas, in a completely random initial state, gradually evolves into a regular crystal-like array of molecules. [The intermolecular potential in this case is taken to be that shown in Figure 2(b)].

RUTHERFORD SCATTERING

The computer generates animated pictures of charge particles having randomly chosen impact parameters, being scattered by a fixed charge particle. The student first needs to specify the radius and charge of the scatterer and the velocity of the incoming particles that he wishes to study. The student can carry out quantitative observations as well as qualitative observations of the families of trajectories resulting from various energy projectiles and different kinds of scatterers (Figure 4). For example, the student can observe the maximum energy projectile for which 180° scattering can occur for different size scatterers having the same charge. He can also observe the relation

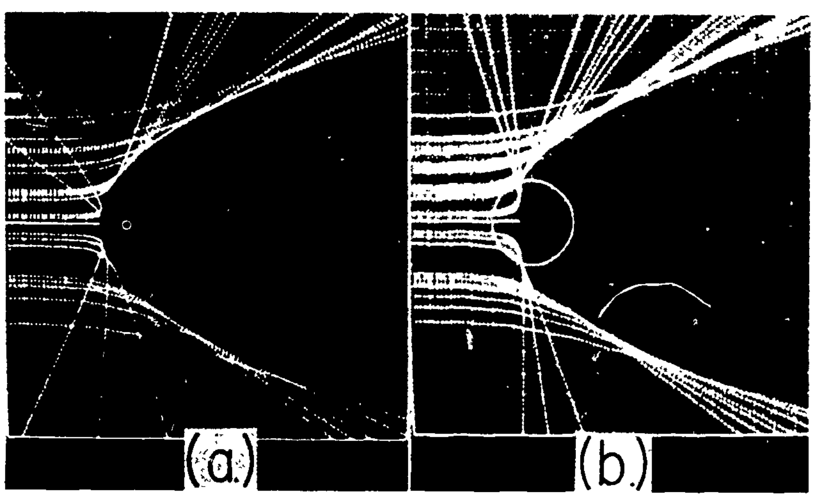


FIGURE 4 Rutherford scattering from (a) a point charge, (b) a uniform sphere of charge.

between impact parameter and scattering angle, from which he can obtain the differential cross section for different size scatterers and force laws.

SQUARE-WELL WAVEFUNCTIONS

Using a technique similar to that reported by Luehrmann,[2] the student can empirically find the eigenfunctions and eigenvalues for a square-well potential. The student first specified the shape of the potential well, and then a trial energy, E. The computer then calculates and displays ψ_E, the corre-

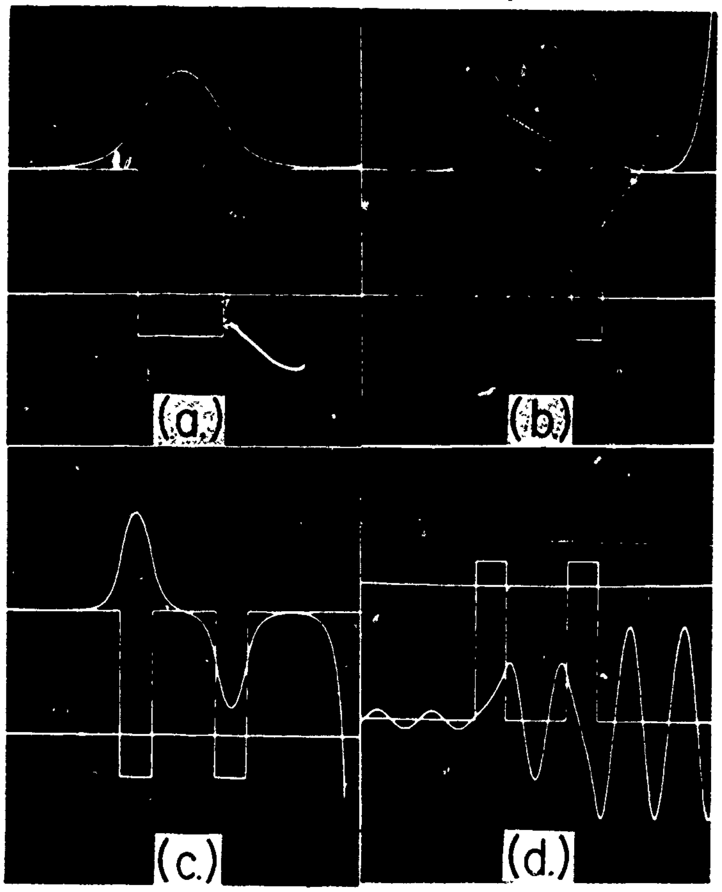


FIGURE 5 Wavefunctions which are nearly eigenfunctions for (a) lowest energy state for a square well, (b) lowest energy symmetric state for a double well, (c) lowest energy antisymmetric state for a double well, (d) the real part of the wavefunction for a double barrier, with the plane wave incident from the right. The horizontal lines show the energy levels.

sponding wavefunction. ψ_E vanishes for large negative x and diverges for large positive x, unless the trial energy is an eigenvalue. The student types in various trial energies, and finds approximate values for the eigenvalues, using the fact that the n^{th} eigenvalue lies between $E + \Delta E$ if the wavefunctions ψ_E and $\psi_{E + \Delta E}$, corresponding to these energies, have n and $n-1$ nodes, respectively.

In addition to the square well, the student can investigate any potential that he can specify in terms of five constant levels, including, for example, the double well, the single barrier, and the double barrier. Once he finds the eigenvalues for a double well, consisting of two widely separated wells identical to the original single well, he then observes the near degeneracy of symmetric and antisymmetric wavefunctions (Figure 5).

The case of the double barrier is also of considerable interest. For each positive energy the student types, the computer calculates the transmission coefficient for a plane wave incident on the double barrier. If the student types a range of energies, and plots the computed transmission coefficient against energy, he observes a series of resonance peaks (Figure 6). The peaks occur at energies for which the distance between the two barriers is a half-integral multiple of the free particle wavelength.

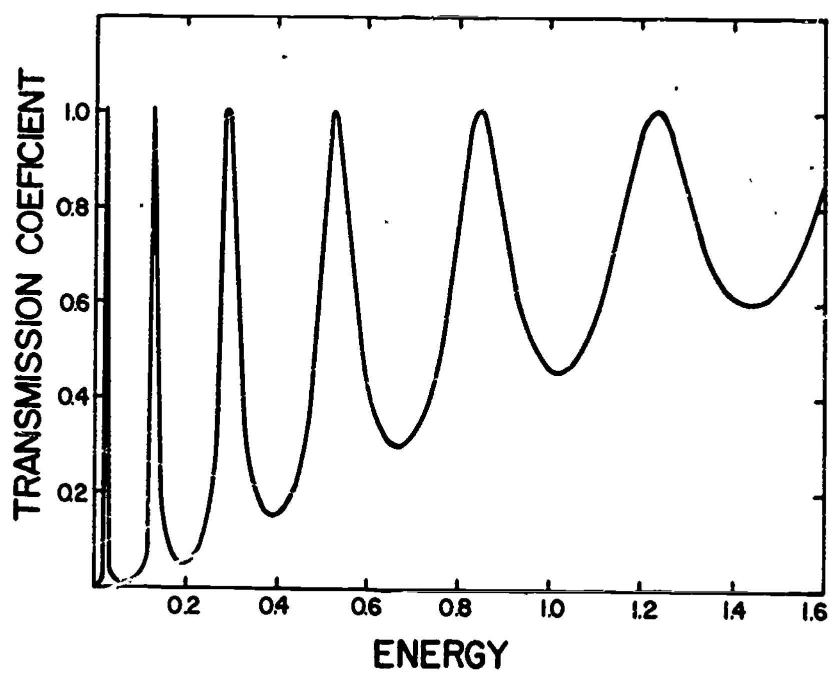


FIGURE 6 Transmission coefficient against energy for a double barrier. $E = 1$ corresponds to the height of the barrier.

INTERFERENCE AND DIFFRACTION

The student can carry out simulated experiments, in which he can study interference and diffraction patterns arising from arrays of slits illuminated by a monochromatic source. Once the student types the wavelength, number of slits, slit width, and slit separation he wishes to study, the computer then generates individual photon hits on the screen, which gradually accumulate to produce an interference pattern like the one shown in Figure 7. This pattern took less than one minute to create on a PDP-6. The particular algorithm that is used to create the display, makes this time roughly proportional to the number of slits. The fact that the student observes both the arrival of individual photons, along with a cumulative interference pattern, serves to illustrate clearly both the wave and particle aspects of light.

The student is urged to investigate whatever slit arrangements he wishes, and to observe at what values of sinθ the significant features (primary and

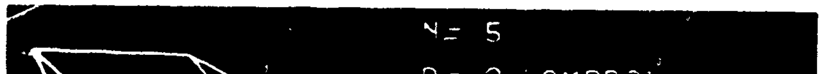


FIGURE 7 Interference pattern resulting from five slits, each of width 10,000 A, separated by 15,000 A, and illuminated by light of wavelength 5000 A.

secondary maxima and minima) of the interference pattern occur. To facilitate these observations, the screen on which the patterns appear is graduated in intervals of $\sin\theta$, covering the entire region from -1 to $+1$. (The program, as now written, assumes the distance from the source to the slits, and from the slits to the illuminated screen, to be infinite.)

The technique by which the computer generates point plots, such as the one in Figure 7, consists of the following steps:

1. Divide the display area into a large number of thin horizontal rectangular strips. (250 is the number currently used.)
2. For each strip, calculate the resultant light intensity I, falling on the strip, normalized to unit intensity for the strip on which the light intensity is greatest.
3. Generate three pseudo-random numbers[3] r_x, r_y, and r from a flat probability distribution ranging from zero to one.
4. Compare the value of the normalized light intensity, I, falling on the strip whose y coordinate is r_y, with the value of r. If $I < r$, then go back to step 3. If $I > r$, then plot a single point whose coordinates are r_x and r_y. (The rectangular display area is defined by $0 < x < 1$, and $0 < y < 1$.)
5. If less than some specified number of points have been plotted, go back to step 3.

HYDROGEN ATOM WAVEFUNCTIONS AND STANDING WAVES ON A DRUM

Hydrogen atom wavefunctions[4] are calculated and displayed (Figure 8) by

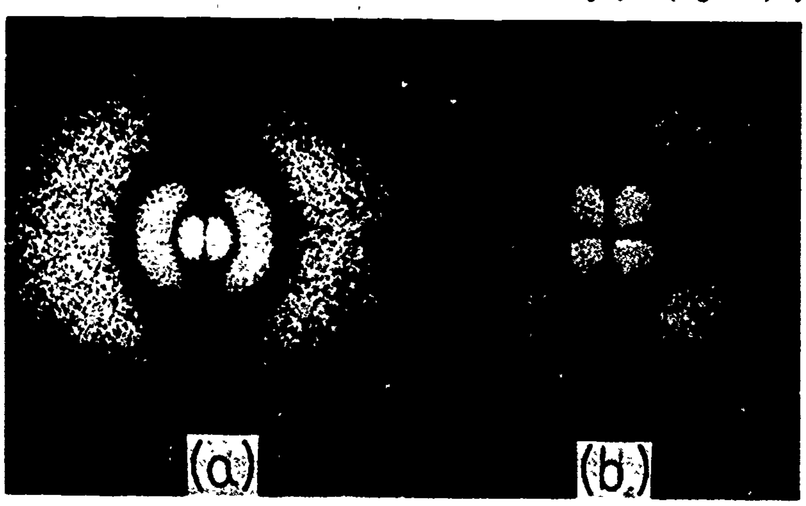


FIGURE 8 Hydrogen atom wavefunctions for specified quantum numbers (n,ℓ,m): (a) 4,1,1, (b) 4,2,1.

showing two-dimensional cross sections (a slab constant ϕ), of the electron probability density, $|\psi|^2$. In the current version of the program, the student can select any quantum numbers subject to the conditions' $n < 5$, $\ell < n$, and $|m| < \ell$.

The technique used by the computer to generate plots of this kind is somewhat different from that described from the previous simulation. The technique used in this case consists of the following steps:

1. Divide the circular display area into a large number of cells using a polar coordinate grid.
2. For each cell, calculate the quantity $x = c|\psi|^2 \Delta A$, where c is some specified constant and ΔA is the area of the cell.
3. Display N points at random positions inside the cell, where N is the greatest integer less than x.
4. Generate a random number r, and if $x - N > r$, then display one more point at a random location within the cell.
5. If more cells remain to be considered, go to step 2.

The technique just described is also used to display standing waves on a drum.

The amplitude of standing waves on a freely vibrating uniform circular membrane of radius R, is given by[5]:

$$a_{mn} = \cos(m\theta) J_m (2\pi r/\lambda_{mn}) \qquad (3)$$

where λ_{mn} are the allowed wavelengths satisfying the boundary conditions $J_m (2\pi R/\lambda_{mn}) = 0$, J_m is a cylindrical Bessel Function, and m and n are

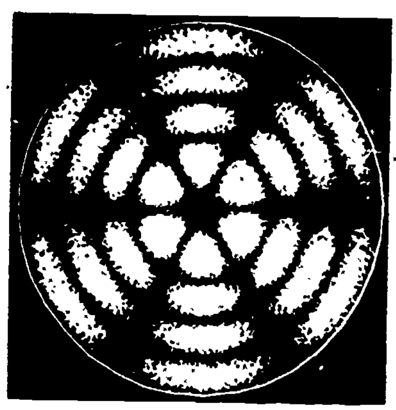


FIGURE 9 Standing-wave pattern on a freely vibrating circular membrane, for the case $m = 3$ and $n = 4$.

integers. The student can type in values of m and n and the computer generates standing wave patterns, such as the one in Figure 9, in which the number of dots per unit area corresponds to the square of the wave amplitude given by Equation (3). (The square was chosen for consistency with preceding examples.) While this procedure is not particularly suitable for on-line simulated experiments, it has been used in the preparation of slides for use as visual aids in physics instruction and for textbook illustrations.[6]

APPENDIX: INFORMATION FOR PROSPECTIVE USERS

The programs described in this article have been implemented so far only on a Digital Equipment PDP-6, which has a 2-μsec memory cycle time. In their present form, the entire set of programs can be loaded in 17,000 words of core. They could be easily separated and run in probably no more than 10,000 words of core. I can give no estimates of CPU time, since all the displays are interactive, and therefore the time depends on the values of parameters that the user specified. All programs, except for some assembly language subroutines used to display points, lines, and characters, are in FORTRAN IV.

Users at other locations could, with some effort, make use of these programs. The effort required would be minimal if used on a PDP-6 or PDP-10. On another machine with a FORTRAN IV compiler, at least 10,000 words of core and a storage display device (such as the Tektronix 611), with FORTRAN-callable subroutines for displaying points, lines, and characters, the amount of programming effort would not be very great. Interested users may obtain, free of cost, program listings and copies of instructions given to each student user, and answers to specific questions. The documentation for the programs is, unfortunately, rather minimal at this moment. I am currently involved in developing a largely new set of simulations for an 8000-word IBM 1130 with no display terminal, which will be much more machine-independent and much better documented. However, I cannot at this moment say when, or under what conditions, this new set will be ready for distribution.

REFERENCES

1. P. R. Benyon, *Simulation 11*, 219 (1968).
2. A. Luehrmann, *Am. J. Phys. 35*, 275 (1967).
3. A. Ralston and H.S. Wilf, *Mathematical Methods for Digital Computers*, John Wiley and Sons, New York, 1964, p. 253.
4. L. Pauling and E. Wilson, *Introduction to Quantum Mechanics*, McGraw-Hill, New York, 1935, p. 134.
5. P. M. Morse, *Vibration and Sound*, McGraw-Hill, New York, 1948, p. 153.
6. H. Y. Carr and R. T. Weidner, *Physics From the Ground Up*, McGraw-Hill, New York, 1970.

Simulation of Optics Experiments

MEYER KATZPER

INTRODUCTION

Some of the earliest educational applications of the computer involved the use of computer programs for the analysis of data obtained from optical experiments. These applications included the demonstration of various lens aberrations and their corrections[1] and also the study of prisms.[2] The capabilities of the computer to simulate optics experiments, however, were not fully exploited.

The flexibility inherent in computer simulation of optics experiments can be used to aid in clarifying basic concepts and to impart a feeling for the relationship between the underlying physical reality and its mathematical formulation. A multiplicity of rays can be traced through any optical system giving an excellent representation of what happens to the light from the viewpoint of geometrical optics. Once the computer takes care of the tedious job of ray tracing, it becomes much easier to study many different optical configurations. The results obtained can be compared with those calculated using mathematical formulations which were created to avoid the process of ray tracing. The simulated optical experiments can be varied to include a range of possibilities not accessible in an ordinary laboratory. Thus, a single form of aberration can be isolated and studied for its effects alone.

A number of optics experiments simulated by ray tracing with a computer are here presented as an example of what can be done. The experimental variables are manipulated by selection of data options. The computer program used provides a series of mathematical surfaces to represent lenses, and it traces rays through each surface following the path of light through the system. The program forms a basis for exercises designed so that to carry out the simulations properly, the student will have to think through the planning, analysis, and interpretation of experiments. The range of application of the various optical relationships is determined by comparison with the ray tracing results representing the experiment.

College of Arts and Sciences, State University of New York, Plattsburgh, New York. The programming work on which this paper is based was carried out at the National Biomedical Research Foundation with the aid of Menfar R. Shiu.

THE PROGRAM

Use is made of a set of programs that allow a choice among a series of options involving the input, output, number of lenses, and their configuration. A small set of data cards is used to specify the desired options. The input consists of either parallel rays or point sources at specified positions. Circular stops are defined by their positions and radii. The lens configuration is given for spherical lenses by the radii, the centers of curvature, and the associated indices of refraction. This allows for the designation of an appropriate lens configuration for any given experiment. The outputs include a saggital ray-tracing diagram, spot diagrams, the Gaussian parameters of the lens system, image locations, least-confusion disk position and size, and spherical aberration. We thus have a versatile setup for carrying out a series of experiments.

A sample data deck for carrying out ray tracing is shown in Figure 1. The cards are identified by the letters on the left for purposes of explanation. The A cards are standard control cards for the IBM 360/44.

The B card is all that is needed to call for the execution of the program called RAYS. A number of programs can be stored in the computer which provide for different output options. A program called DENSITY will provide a plot of the energy density at any given plane. The program we are illustrating can provide the following output:

(1) paraxial focal position;
(2) least-confusion disk position;
(3) least-confusion disk radius;
(4) Gaussian optical parameters of the system;
(5) plot of the ray tracing for the system;
(6) the spherical aberration for different aperture sizes.

```
A   //      JOB
A   // SYS004 ACCESS TAPE, 181=
B   //  EXEC RAYS
C           PARALLEL RAYS THROUGH 4 SURFACES
D   RAYS    PL    4          150   -1    2
E   LENS       1.0        1.376     1.336     1.41      1.0
F   LENS  1     0.0        0.0       7.7       7.7       7.7     7.7
F   LENS  2     0.0        0.0       7.3       6.8       6.8     6.8
F   LENS  3     0.0        0.0      13.6      10.0      10.0    10.0
F   LENS  4     0.0        0.0       1.2       6.0       6.0     6.0
G   LENS  1.7       1   0.0
A   /*
A   /&
```

Identifying letters for explanation

FIGURE 1 Control cards and sample data deck for program RAYS.

The C card allows any title to be entered to label the output. The D card illustrated specifies that ray-tracing output is desired (PL), the number of lens surfaces (4), the number of input rays (150), a parallel ray source (−1), and Gaussian optics results (2). Card E gives the indices of refraction. The F cards specify the geometry of the lens system. Their general form is.

NAME N CX CY CZ RX RY RZ

NAME is an arbitrary four-letter label; the surface number is given by N. The origin of the radius of curvature is specified by CX, CY, CZ. The distances are measured from the vertex of the first surface with respect to the incoming light. The magnitude of the radius of curvature is determined from RX, RY, and RZ. Card G specifies the aperture radius, the number of stops of interest, and the location of the stop.

APPLICATIONS

To within the validity of the assumptions of geometrical optics, ray tracing can accurately simulate a large range of laboratory experiments. The examples that follow illustrate the application of the simulation approach.

1. The refraction of light is studied by having parallel rays incident upon curved surfaces with different indices of refraction. The use of variations in density of the traced rays to represent changes in light intensity is thus made clear.

2. Rays incident, through a small aperture, upon a series of spherical surfaces with different radii of curvature will give data on the change of focal position as a function of curvature. The student is then instructed to study the effects of variation of the aperture size. The ray-tracing results thus obtained will show the increasing effect of spherical aberration with aperture size and the resulting difference between the least-confusion disk focal position and the paraxial focal position (see Figure 2).

3. The first study of lenses takes up the thin lens equation;

$$\frac{1}{o} + \frac{1}{i} = \frac{1}{f}$$

where o is the object distance, i the image distance, and f the focal length all measured from the center of the lens. The value of f is given by

$$\frac{1}{f} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right)$$

where n is the index of refraction of the glass and R_1 and R_2 are the radii of curvature of the lens.

It is evident from this equation that the shape of the lens may vary and still maintain a fixed focal length. These different lenses are characterized by their shape factor, which is given by

$$Q = (R_2 + R_1)/(R_2 - R_1)$$

A given set of lenses having the same aperture and paraxial focal length but different shape factors will have differing amounts of aberration so that the student can find the test shape factor for which the aberration is a minimum.

4. A check on the range of validity of the thin-lens equation for paraxial rays can be carried out by increasing the thickness of the lens in small steps. At each step the results can be compared with those obtained from ray tracing.

5. For thick lenses and for lens combinations the concept of principal planes can be introduced. The calculations will be carried out by the program to demonstrate that Gaussian optics must be used in cases where the thin-lens equations are no longer applicable. The breakdown of Gaussian optics results due to factors such as spherical aberration are studied by increasing the aperture size. The saggital ray-tracing output (Figure 2) serves as a basis for measurements. The spot diagram output which illustrates the variation of

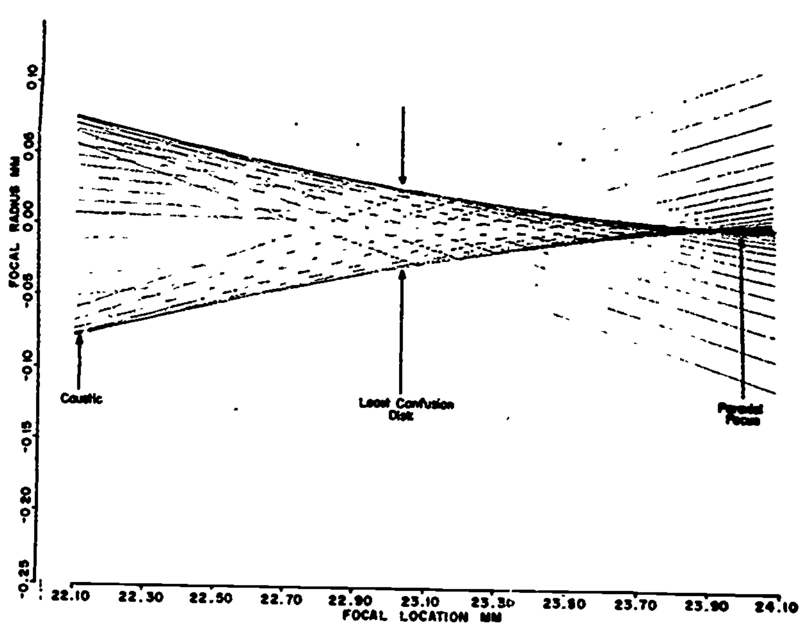


FIGURE 2 Enlargement of ray-tracing results near the focal plane for a saggital cross section of a series of spherically symmetric centered lenses.

light intensity in different planes is easily measureable (Figure 3). Similarly resolution from the point of view of geometrical optics is studied (Figure 4) by examining the images of two point sources.

6. The computer can be involved in problem solving and the determination of unknowns. A coded set of unknowns are incorporated in the program so that the student must discover the correct values by using a minimum number of test inputs. Given the radii of curvature of a thin lens with an unknown index of refraction assigned to it in the program, a student must first use the program to find the focal position. Only after completing this step can he solve for the index of refraction.

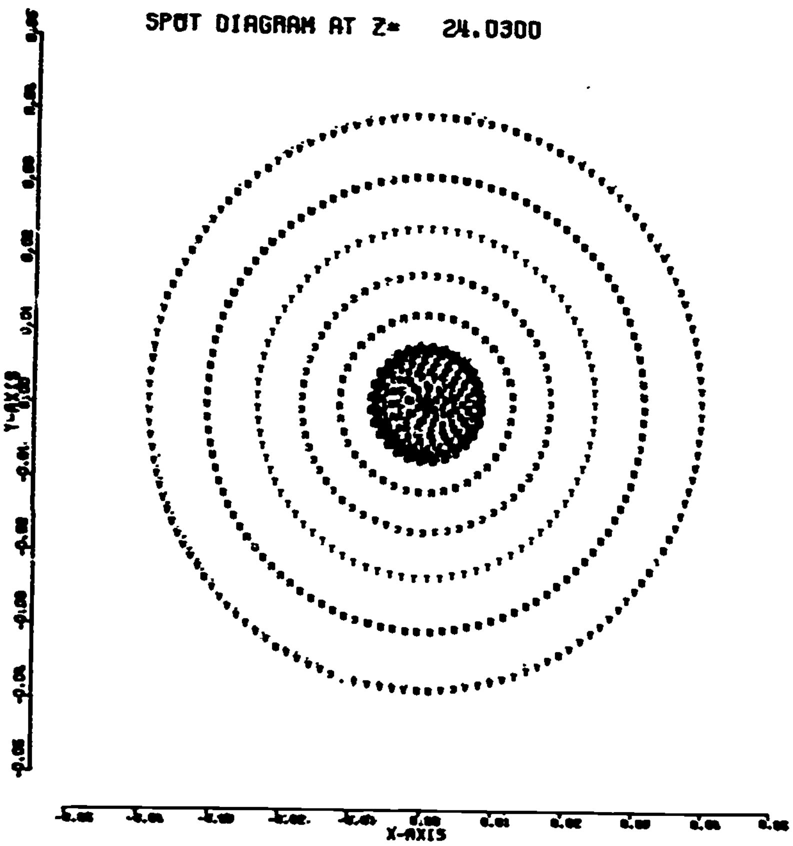


FIGURE 3 Spot diagram at the least-confusion disk location for a lens system representing the eye. Distance of spot diagram from vertex of cornea = 24.03 mm. Different letters represent input rays incident at differing radial distances from the axis and indicate the change in the relative positions of the rays at various planes.

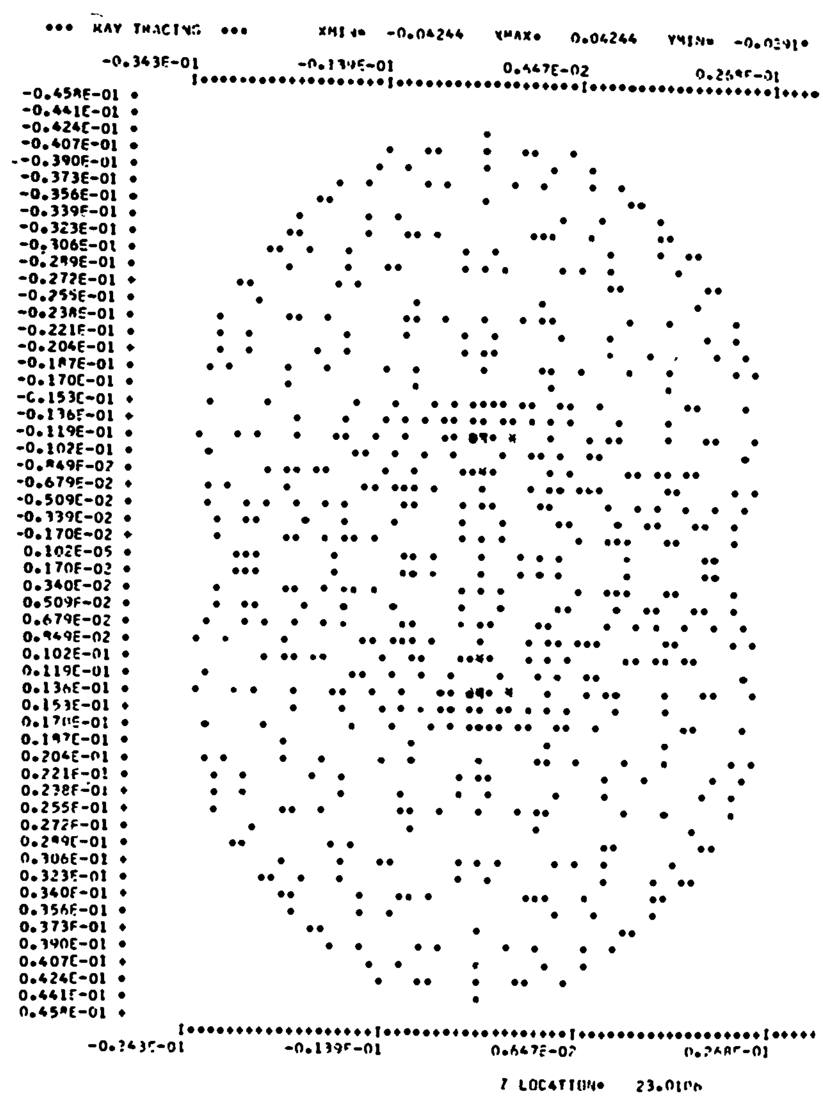


FIGURE 4 Spot diagram illustrating the resolution of two point sources as computed using only geometrical optics. The sources were off the optical axis and separated by 0.02°. The rays were taken through an optical system representing the eye.

The programs used for the simulation of optics experiments described in this paper have been implemented on the IBM System/360 Model 44. The time required for the simulation varied from seconds for tracing a small number of rays through the system to get the paraxial focal position to over an hour for tracing 1000 rays through many surfaces to produce spot diagrams at a large number of positions.

It is not feasible to perform these simulations on the computer presently installed on the Plattsburgh Campus, but it is planned to convert these simulations for interactive use by students on the Burroughs B-3500 (to be installed at Plattsburgh in 1971) and the IBM System/360 Model 67. This is expected to be completed during 1971. When this is complete, the simulations will be subjected to extensive use by students and faculty. Listings and other documentary material will be made available after the experience at Plattsburgh has been documented.

REFERENCES

1. A. B. Grossberg, *Am. J. Phys. 37*, 559 (1969).
2. R. D. Hartmann, *Am. J. Phys. 34*, 793 (1966).

Thermodynamics of Ising Lattices. A Computer Simulation of a Phase Transition

SAMUEL WILEY

INTRODUCTION

With the advent of large-memory, high-speed computers a number of problems in statistical mechanics have been approached through the simulation of their thermodynamic behavior using the Monte Carlo technique. This method has been used as a research tool to investigate the equation of state for certain models of gases[1] and to calculate the thermodynamic behavior of binary alloys[2] and magnetic lattices.[3] This paper presents an adaptation of this technique for use in upper-division courses on thermodynamics and statistical physics.

The general approach used in these simulations is quite straightforward. A sequence of states is generated for the system under study in such a way that the frequency of occurrence of each state is proportional to the Boltzmann distribution factor $\exp(-E/kT)$, where k is the Boltzmann constant, T the temperature of the system and E is the energy corresponding to the particular state. The thermodynamic properties of interest are then obtained by taking their average values over the sequence of states so generated.

The straightforward nature of the calculations provides a very good picture of the microscopic character of statistical systems and gives the student insights into their physical behavior which are difficult to obtain in the more conventional analytical treatment in which the physics may be obscured by the mathematical calculations required. This is particularly true of the phenomenon of phase transitions which can rarely be approached on the undergraduate level due to the difficulty of the analysis required.

THE ISING PROBLEM

The Ising model was chosen for this discussion because it is a simple model of

Department of Physics, California State College, Dominguez Hills, California.

a magnetic lattice which has been shown, in the two-dimensional case, to exhibit a phase transition. A scalar coordinate is associated with each magnetic spin site in the lattice and is allowed only two possible values, one corresponding to an "up" spin at that site, the other representing a "down" spin at the site. All effects other than a spin-spin interaction and the interaction of the spins with an external magnetic field are ignored.

The interaction energy between the spins at the jth and kth sites of an Ising lattice is given by

$$V_{jk} = -J_{jk} s_j s_k \tag{1}$$

where J_{jk} is the coupling constant between these two sites and

$$s_j = \begin{cases} 1 & \text{if the } j^{th} \text{ site contains an up spin} \\ -1 & \text{if the } j^{th} \text{ site contains a down spin} \end{cases}$$

Obviously $J_{jk} > 0$ yields a lower energy for parallel spin states and thus represents ferromagnetic behavior, while $J_{jk} < 0$ corresponds to the antiferromagnetic case. The total energy of the Ising system in the absence of an external magnetic field is then

$$E = -\sum_{j<k} J_{jk} s_j s_k \tag{2}$$

In this treatment of the Ising problem, only nearest-neighbor (nn) interactions will be considered and the coupling constant will be assumed to be positive and uniform for all sites so that

$$J_{jk} = \begin{cases} J & j, k \text{ nearest neighbors} \\ 0 & \text{otherwise} \end{cases}$$

Equation (2) then becomes

$$E = -J \sum_{\langle nn \rangle} s_j s_k \tag{3}$$

where the sum is over nearest neighbors only.

If, in addition to this spin-spin interaction, the spins also interact with an external magnetic field, B, and if the "up" spin direction is chosen to coincide with the direction of the magnetic field, then the total energy becomes

$$E = -J \sum_{\langle nn \rangle} s_j s_k - B \sum_j s_j \tag{4}$$

Ising[5] solved the one-dimensional chain with nearest-neighbor interactions for arbitrary values of an applied field, but the solution exhibited no phase transition. Of more interest, Onsager[6] solved the two-dimensional zero field case using a rather ingenious (and difficult) mathematical method. Following Onsager's paper, simpler derivations were published by Kaufman,[7] using spinor notation, and Kac and Ward,[8] based on a combinatorial formulation of the problem. These results show that the two-dimensional infinite lattice exhibits a singularity in the specific heat vs. temperature curve. This is one of the few mathematical models for which an exact solution is available that exhibits a phase transition. The location of the specific heat singularity corresponds to the Curie Temperature of the lattice. Unfortunately, the more general non-zero field case and the three-dimensional lattice have never been solved rigorously.

COMPUTER SIMULATION OF THE THERMODYNAMIC BEHAVIOR

The programs used in this simulation are designed to process linear, square, or cubic Ising lattices with nearest-neighbor interactions in an arbitrary external field. This variation in the dimensions of the lattice is of particular interest since, as has been pointed out, the one-dimensional case exhibits no phase transition in contrast to the two-dimensional lattice, while the three-dimensional problem has never been solved exactly. In this way the student can observe the dependence of the phase transition on the dimensionality of the situation, progressing from a problem that is thermodynamically unexciting to one that is still of current research interest. In addition, the size of the lattice and the strength of the external field can be varied so that the effect of these parameters can be observed.

The calculational technique is an adaptation of that utilized by Yang,[3] extended to include the non-zero external field case. The ith configuration of a lattice containing N sites is completely described by the N-component vector $\mathbf{s}(i)$ whose components are the values of the N spin coordinates, s_j, corresponding to the particular configuration. Since s_j can have the value of either +1 or −1, there are 2^N possible configurations for the lattice. To sample the energy distribution of these configurations, a machine simulation of the physical approach to a statistical equilibrium distribution is used. Starting from an arbitrary spin configuration, a sequence of states is generated by processing each lattice site, one at a time. At each site the consequence of flipping the spin at that location (leaving all other spins unchanged) is examined and the probability of this spin remaining unchanged is set equal to the appropriate Boltzmann factor.

The probability of the spin remaining unchanged is then given by

$$P = \frac{\exp(-E/kT)}{\exp(-E/kT) + \exp(-E'/kT)}$$

E = energy before flip

E' = energy after flip

This value is compared to a computer-generated random number between zero and one. If P is less than or equal to the random number the spin will be flipped, otherwise it will remain unchanged. After all the sites have been processed in this manner, one cycle of computation has been completed and the program returns to the first site to start a new cycle. At the end of each cycle the total energy $E(i)$ and long-range order $M(i)$ (average spin per site) are computed and stored. When a predetermined number of cycles has been completed, expectation values of energy and long-range order are obtained by averaging the values obtained at the end of each cycle of computation.

$$\langle E\rangle = \frac{1}{n}\sum_i E(i)$$

$$\langle M\rangle = \frac{1}{n}\sum_i M(i) \tag{6}$$

where n = total number of cycles of computation. Further, the specific heat is given in terms of the energy by

$$C = \frac{\partial E}{\partial T} \tag{7}$$

It is not easy to obtain this differential accurately from the discrete set of energy values generated, but Yang has shown that the specific heat can be expressed exactly in terms of the energies as

$$C = (1/T^2)\left\{\left[\frac{\Sigma(E(i))^2}{n} - \sum\left(\frac{E(i)}{n}\right)^2\right]\frac{n}{n-1}\right\} \tag{8}$$

Thus all the thermodynamic quantities of interest are obtained from the energies $E(i)$ and long-range order values $M(i)$ of the final states of each cycle of calculation.

SUMMARY AND SAMPLE RESULTS

In practice this simulation problem is usually presented to select students in the class. They are responsible for the investigation of the effect of certain parameters and the discussion of their results before the rest of the class. (Some instructors might prefer to generate the results themselves and present them to the class for discussion.) These students are supplied with a description of the problem, much like the one given here, and source decks for the

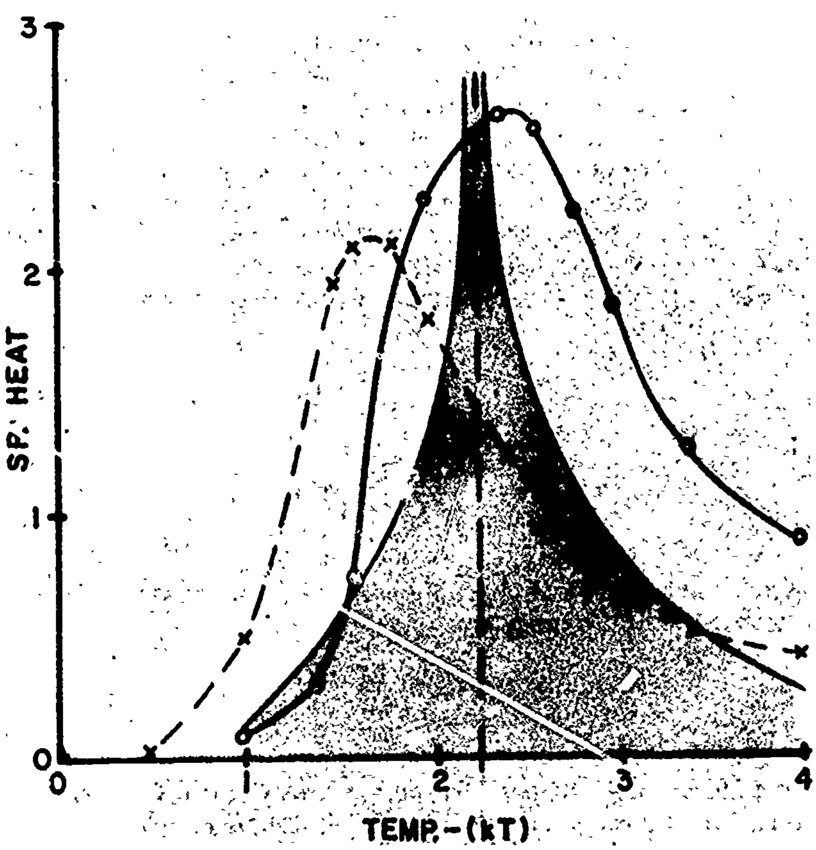


FIGURE 1 Specific heat curves vs. temperature for a 4 X 4 lattice with periodic (solid curve) and open (broken curve) boundary conditions. Onsager's result is shown by the shaded outline, the broken vertical line gives the theoretical value of the Curie Temperature. All values were obtained for $J = 1$.

programs. Parameters that can be varied are lattice dimension, lattice size, boundary conditions (periodic or open), number of cycles of computation, applied magnetic field and temperature. Some examples of the effects that can be illustrated follow.

To demonstrate the effect of the size of the lattice and the effect of the boundary conditions, the specific heat vs. temperature curves are shown for a 4 X 4 lattice and a 12 X 12 lattice in Figures 1 and 2. These results are plotted for zero external field and were obtained after 500 cycles of computation. The specific heat curves were calculated for both open and periodic boundary conditions. For comparison, Onsager's result for the critical temperature of an infinite two-dimensional lattice is also shown.

The magnetic-nonmagnetic transition is characterized by an abrupt change from an ordered to a disordered state as the temperature of the lattice is

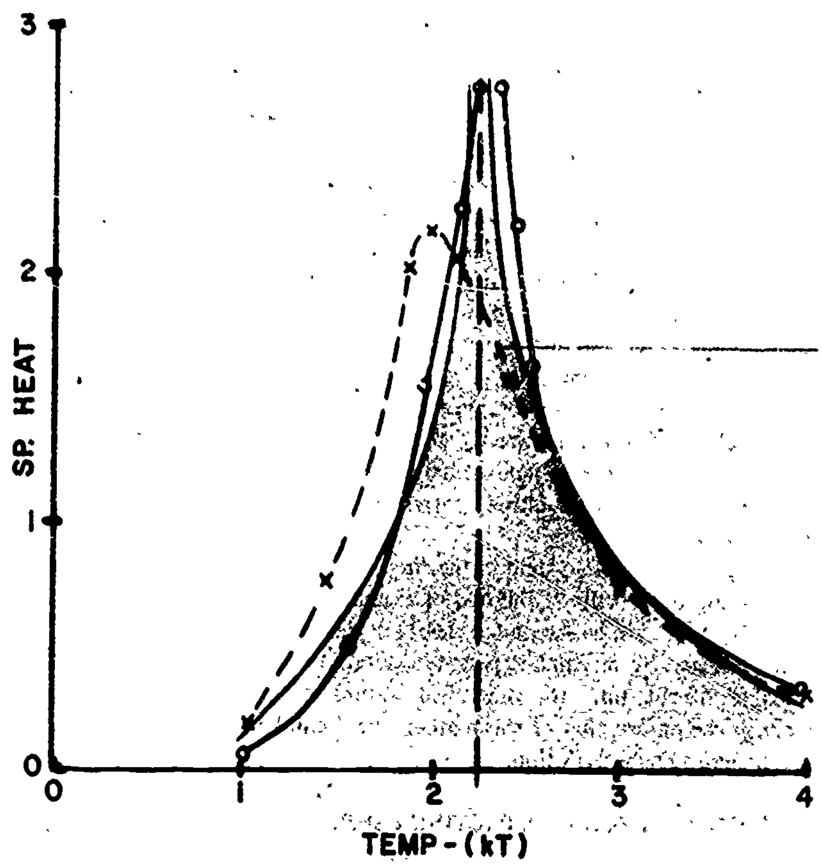


FIGURE 2 The same as Figure 1, but for a 12 X 12 lattice.

increased. The value of the temperature at which this transition occurs is known as the Curie Temperature. This change of state is accompanied by a singularity in the specific heat, as can be seen from the plot of Onsager's results. No finite system will duplicate this singularity, but note that in every case the specific heat curves have fairly sharp peaks and that the peaks for the open lattice (the broken line) and the periodic lattice (the solid line) bracket the value of the Curie Temperature obtained by Onsager. The peak for the periodic boundary condition occurs at a temperature above this value, and that for the open boundary condition is below the Curie Temperature. This is to be expected from intuitive arguments since the periodic lattice introduces a stronger correlation between spins than does the open lattice, but as the lattice size increases the distinction between the different boundary conditions disappears. Thus as the lattice size is increased, a better approximation to the behavior of the infinite lattice is obtained, while at the same time the boundary effects have less influence so that the two specific heat curve peaks bracket the Curie Temperature more closely. This effect can be seen by comparing Figures 1 and 2. The value of the Curie Temperature could then be isolated to any degree of accuracy desired by simply increasing the size of the lattice. However, as can be seen from the results for the 12 X 12 lattice (Figure 2), good accuracy can be obtained for relatively small latice sizes.

For comparison with the two-dimensional results, specific heat vs. temperature curves for an 8 X 8 X 8 lattice with periodic boundary conditions are shown in Figure 3. The similarity between these curves and those obtained for the two-dimensional lattices gives a strong indication that the three-dimensional lattice also undergoes a phase transition.

The effect of the application of an external magnetic field is also shown in Figure 3. The solid line represents the result for zero field, while the broken line is obtained for a non-zero value of the external field. As would be expected, the presence of an external field shifts the peak of the specific heat curve to a higher temperature, corresponding to an increase in the Curie Temperature.

The approach of the system to thermodynamic equilibrium can be observed by printing out results at the end of regular intervals of cycles of computation. The effect can be seen most clearly in the long-range order values.

In summary, by varying the parameters indicated above, the student can get valuable insight into the statistical nature of thermodynamic behavior through the observation of this mathematical simulation of a physical system undergoing a phase transition.

INFORMATION FOR PROSPECTIVE USERS

The program described in this paper is currently operating on a CDC 3300 with 84K of core. This computer is shared by the six California State Colleges

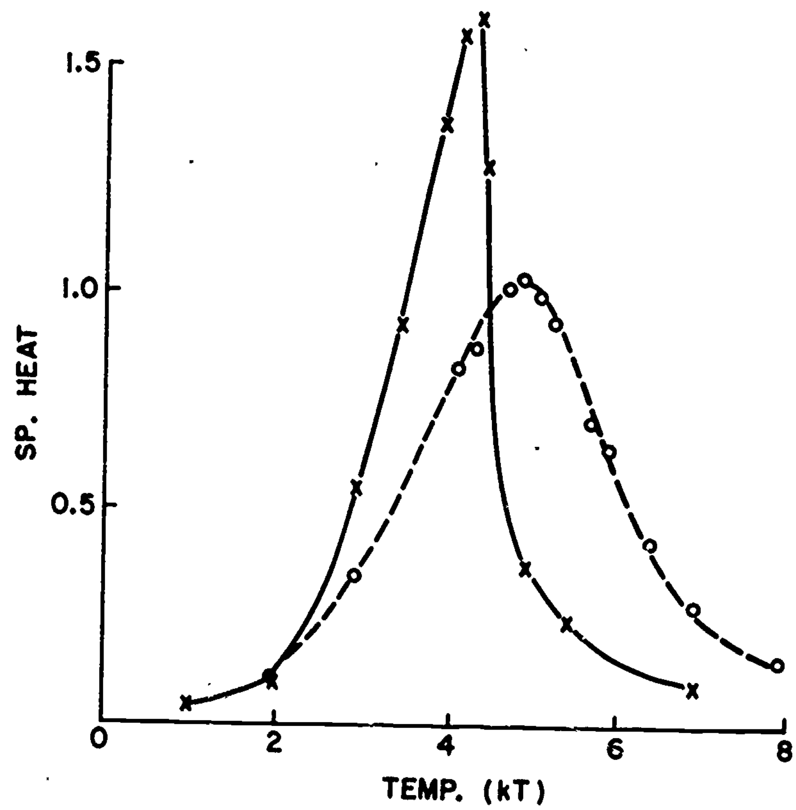


FIGURE 3 Specific heat vs. temperature for a three-dimensional 8 × 8 × 8 lattice. The solid line is the zero-field result, the broken line is for a non-zero field value, $B/J = 0.5$

in the Los Angeles area by means of remote entry terminals. Because of the input-output requirements of this system, the maximum usable core for any one program is 32K. Core limitations, of course, place restrictions on the maximum lattice size and the maximum number of cycles of computation that can be used in the calculations. The program could be implemented on any facility having appropriate programming languages and core size. The programming languages used are FORTRAN IV for the main program and COMPASS for subroutines.

Computation time, is, of course, dependent on the lattice size and the number of cycles computed. For the 8 × 8 × 8 lattice with 500 cycles of computation, each data point required approximately three minutes of CPU time. This corresponds to 256,000 individual lattice site calculations, so times for lattices of differing size may be estimated from this information.

Complete program listings are available from the author on request. These may require slight programming adjustments from system to system to allow for local variations in FORTRAN languages and in assembler language for the subroutines.

REFERENCES

1. D. Metropolis *et al., J. Chem. Phys. 21*, 1087 (1953).
   M. N. Rosenbluth and A. W. Rosenbluth, *J. Chem. Phys. 22*, 881 (1954).
   W. W. Wood and F. R. Parker, *J. Chem. Phys. 27*, 720 (1957).
   W. W. Wood and J. B. Jacobson, *J. Chem Phys. 27*, 1207 (1957).
2. L. D. Fosdick, *Phys. Rev. 116*, 565 (1959).
3. C. P. Yang. *Proceedings of Symposia in Applied Mathematics XV*, 351 (1963).
4. G. F. Newell and E. W. Montroll, *Rev. Mod. Phys. 25*, 353 (1953).
   C. Domb, *Phil. Mag. Supplement 9*, 151 (1960).
   T. Schulz, D. Mattis, and E. Lieb, *Rev. Mod. Phys. 36*, 856 (1953).
5. E. Ising, *Z. Physik 31*, 253 (1925).
6. L. Onsager, *Phys. Rev. 65*, 117 (1944).
7. B. Kaufman, *Phys. Rev. 76*, 1232 (1949).
8. M. Kac and J. C. Ward. *Phys. Rev. 88*, 1332 (1952).

History of a Failure in Computer Interactive Instruction

EDWIN F. TAYLOR

This paper deals with an elegant and technically successful computer interactive display that has not influenced many students. Every result of special relativity applicable to one spatial dimension can be illustrated using the interactive computer display described in this article. The display presents the *xt* spacetime diagram and allows the placing of event-points and their Lorentz transformation on command, among many other features. The program is powerful, illuminating and simple to use, yet almost no one does use the display. Since there is no excuse for failure as far as the single tool is concerned, we are free to ask why such an excellent tool is not used, and, by implication, what hard questions are in order concerning any proposed technical aid to education. There is the possibility that the printing press will redeem the computer; study booklets illustrated with computer stills may yet make possible widespread use of these displays after all.

THE DISPLAY

In the summer of 1967 a group of physicists and computer specialists were brought to the MIT Education Research Center to consider what educational uses could be made of the computer facilities available there. Part of this group (see **Acknowledgments**) performed the work which forms the basis of this paper, using a Digital Equipment Corporation PDP-7 computer. This computer has a display console with a standard "light pen." The machine can detect when the pen is pointed at a particular illuminated spot or symbol on the screen and may be programmed to interpret this as a command. By using several or many such "light buttons," an inexperienced operator (student) can give a wide range of instructions to the computer. The rows of labels at the bottom of the spacetime display (Figure 1) are mostly such light buttons. Their use is explained in Table 1.

Education Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts.

TABLE 1 Light Buttons of the Spacetime Display (Refer to Figures 1 and 2)

Symbol	Operation
+	Tracking cross (lower right corner of Figure 1). When the light pen is pointed at the tracking cross and then moved across the face of the display, the tracking cross follows. Coordinates x and t of the position of the cross are continuously displayed below the diagram. Removing the light pen leaves the tracking cross where last placed.
EVENT	After placing the tracking cross, the operator touches the button EVENT. The computer then places an event-point at that location and returns the tracking cross to the lower right corner of the display. Up to 16 points can be displayed simultaneously, each point automatically labeled in alphabetical order. When an existing event-point is touched with the light pen, the x and t coordinates of the event are displayed to three-place accuracy below the diagram.
JOIN	Joins the next two points touched by the light pen with a straight-line segment.
VERT	Draws a vertical construction line through the event-point next touched by the light pen, which remains vertical when the display is Lorentz-transformed.
HORIZ	Draws a horizontal construction line through the event-point next touched by the light pen.
−LT+	Lorentz transformation buttons. When the light pen is held to the plus sign, the relative velocity between frames increases continuously. The x and t axes remain perpendicular to one another, but each event-point moves to its coordinate position measured by a rocket observer traveling at a speed relative to the original frame indicated by v/c in the lower left of the diagram. Associated lines and hyperbolas follow their respective point-events. When the light pen is held on the *minus* sign, the reverse transformation takes place.
LIGHT	Two lines 45 degrees from the vertical are drawn through the next event-point touched after this button. These are world lines of light flashes converging on and diverging from the event.
HYPERB	The next event-point touched has drawn through it its invariant hyperbola with respect to the origin point O. When the Lorentz transformation button is held down, the event-point moves along its own invariant hyperbola.
REFLECT	Reflects all points about the vertical axis. Useful in analyzing one-dimensional collisions.
SHIFT	Moves origin O to a point designated by the tracking cross. All other event-points and constructions move along with it.

Symbol	Operation
OOPS!	Error-correcting feature. Restores display to its condition *before* the last button-push (one-step memory only).
–VEC+	Takes the ordinary vector sum (+) or difference (–) of next two event-points touched. A new event-point appears at the location of this sum or difference. Can also take a "roving" sum or difference of any point and the tracking cross; in this case two new event-points appear when the tracking cross is placed and EVENT button is touched: one at the tracking cross and the second at the sum or difference position.
ERASE	May be used to erase selectively any event-point or any construction. (Example: touching ERASE then JOIN then points A and B would erase only the line joining A and B.)
RESET	Erases all points and lines and resets indicators to zero.
+ZOOM–	Holding this button continuously contracts or expands the scale of the display about the origin O without changing numerical value of coordinates of any event-point. Allows closer look (magnifying) at events near the origin or a wider view (telescoping).
DV/C	Touching this button resets the attached number to zero. Can be used to sum *additional* velocity changes during further Lorentz transformations. May be used to demonstrate that velocities do not add.

Typical displays, as shown in Figures 1 and 2, are animated spacetime diagrams that demonstrate how the x (space) and t (time) coordinates of an event observed in one frame of reference are related to the corresponding x' and t' coordinates of the same event as observed in a second frame of reference in motion with a high uniform velocity along the x-direction with respect to the first frame. The horizontal axis in Figure 1 is the x-axis. The vertical axis is the t (time) axis with time coordinate expressed in units of length by plotting ct. The location of a point in the xt plane (such as point A) gives the space and time coordinates of that event (see coordinate reading for point A in lower right portion of the diagram). Coordinates of events in all frames are measured with respect to some common agreed-upon reference event ("starting gun") taken by definition to occur at the zero of time and space (event 0 at the origin of Figures 1 and 2).

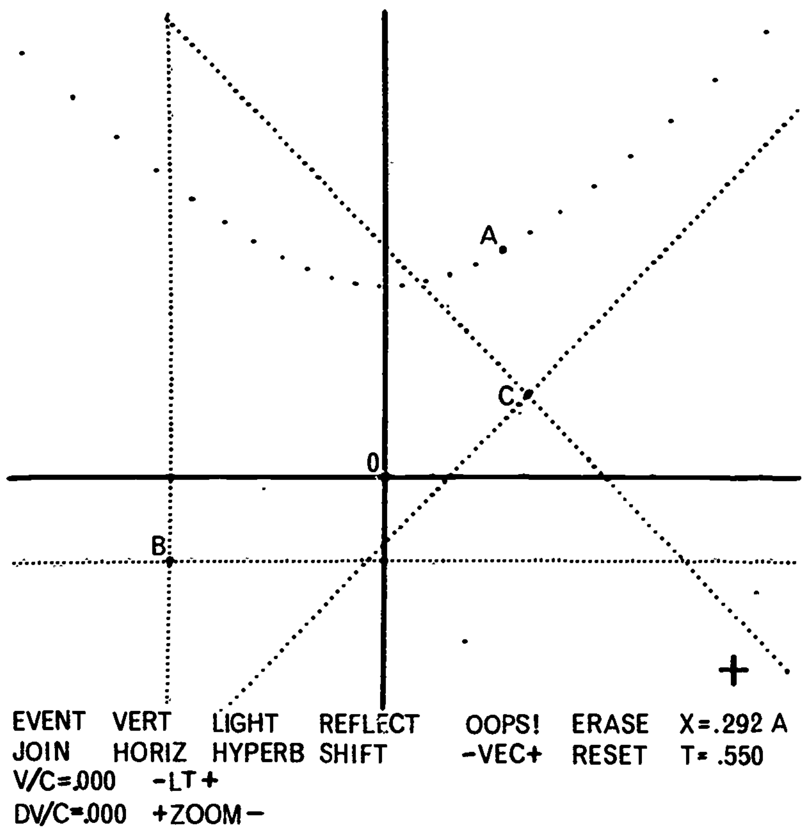


FIGURE 1 Spacetime display, reversed black-for-white. Horizontal and vertical solid lines are x and t axes, respectively. Invariant hyperbola has been drawn through event-point A, horizontal and vertical dotted construction lines through point B, and light lines through point C.

Figure 2 shows the field of events in Figure 1 as measured with respect to a rocket reference frame moving at a speed $v/c = 0.622$ along the x spatial direction with respect to the original reference frame (value v/c is recorded automatically in the lower left corner of the display). The transformation of the coordinate plot is accomplished by touching the light pen to the plus sign in the "Lorentz transformation button" (–LT+) on the display (see Table 1).

Using the light-buttons on the display, one can investigate a wide variety of the properties of spacetime, including: the world lines of particles and light flashes–their paths through spacetime; invariance of the interval $(ct)^2 - x^2$ that separates two events; the light cone as a partition in spacetime (in-

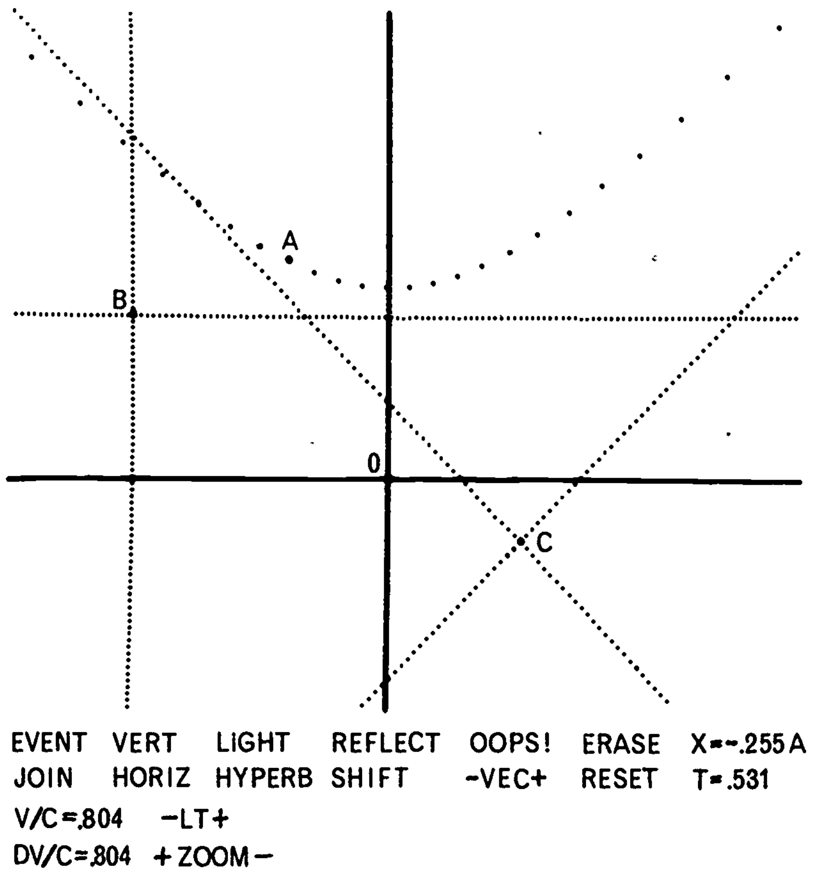


FIGURE 2 Spacetime diagram of Figure 1 transformed to that of rocket observer moving at $v/c = .622$.

variance of the speed of light); regions of spacetime: timelike, lightlike and spacelike relations between events; clocks using light pulses; Doppler shift; time dilation; Lorentz contraction; non-additivity of velocities.

The components of momentum and relativistic energy of a particle transform from one frame to another in the same way that the space and time coordinates of an event transform. This means that a point on the spacetime diagram can be used to represent the x-momentum (horizontal dimension) and energy (vertical dimension) of a particle. Using this interpretation of the diagram, the operator can analyze one-dimensional collisions between particles. In particular the REFLECT button reflects all points about the vertical axis. This can be used to represent the exchange of x-momentum of two particles in an elastic collision as observed in the center-of-momentum frame.

The spacetime diagram is especially versatile in that the student can manipulate it both to get a qualitative feel for relativistic phenomena and also to read numbers from the display accurate to three places. In this way he can pass easily among the quantitative, the analytic, and the intuitive aspects of learning relativity. In summary, every result of special relativity applicable to one spatial dimension can be illustrated on this display.

THE FAILURE

The failure of the spacetime display can be simply described: it is not used by either faculty or students.

After the display was perfected it became apparent very quickly that there was no existing course at MIT in which it could be used interactively by every student individually. Special relativity is taught at MIT in the second semester of a two-year introductory physics sequence. The course is given both fall and spring, typical enrollments being 600 in the spring and 130 in the fall. We saw no practical way for each one of even 130 students to have a significant amount of time scheduled on our single display, particularly since the computer is used for other educational projects as well. Simple-minded duplication of the existing one-display computer would cost $150,000 per display, not including maintenance and supervision. While a small group of students might use the computer as a special experiment, it was too expensive to be applicable to larger numbers.

We still felt, in the flush of technical success, that so beautiful a tool must somehow be able to cure problems of application by sheer virtuosity. So we made a video tape of some manipulations on the display and used this as a large-lecture demonstration. Naturally the interactive feature was entirely lost in this transaction. Two such tapes were made, and both were received with polite indifference. Neither tape was used again.

We next approached the Education Development Center of Newton, Massachusetts, about making a film of the display in operation. However, since the "motion" of event-points on the screen corresponds to the Lorentz transformations of these points to their positions in a moving frame, it was felt that this might, in fact, be confused with the motion of particles in time as observed in everyday life. In short, the display as presented on film might even have negative educational value. Consequently, the professionals at the Education Development Center suggested we produce a picture book of stills that would demonstrate various relativistic effects. A minimum of text material could alternate with questions and answers, possibly in a programmed-instruction format. In this way the development of intuition concerning the Lorentz transformation could be assisted and applied to many classic relativistic effects and one-dimensional collisions.

In sum, the spacetime diagram interactive display mostly sits idle, available instantly on magnetic tape but rarely used. (The computer on which it plays *is* in use for other projects.) Once or twice a month the display is called up for the wonder and delight of a visitor. An interesting set of introductory exercises for the display was written by Wolfgang Rindler and the author. But those students who know about the display have not recognized that many regularly assigned exercises in relativity can be done on it in a few minutes. Or perhaps they feel that the additional inconvenience of scheduling its use on an individual basis is not worth the effort.

QUESTIONS TO ASK ABOUT A PROPOSED TECHNICAL LEARNING AID

A single failure, no matter how sobering, cannot provide a prescription for success. Therefore, instead of listing the lessons we have taken from our failure, we list some questions one might ask about a proposed technical system that is designed to fulfill or assist in fulfilling some educational purpose. Almost without exception we failed to ask these questions about the spacetime display. The list does not include (and should be added to) questions about the professional correctness of the materials, the strategy of their presentation, the resources necessary to develop the tool, the competence of those making the proposal, long-term effects on the curriculum. and the human effects of the resulting innovations.

1. Does this device truly teach *anything*?
2. *What* does it teach?
3. *How well* does it teach compared with already-existing methods or simpler alternative methods that might be developed?
4. Have students been involved in the planning for this device?
5. Have preliminary or mock-up versions been tried with students?
6. How do students respond? (No students were included in the discussions of our display design, nor were preliminary versions or hand-drawn mock-ups tried on them to see what they found useful. Seen in retrospect, the entire operation has the aspect of enthusiastic millionaires in their private club planning an anti-poverty program.)
7. How much of his own money would a student be willing to spend to use the device for one hour? (Ask him!)
8. Are local faculty who might use the device participating in its design and execution?
9. Is it based on an established technology, defined as one for which professional commercial troubleshooting and repair is quickly available locally?

10. If maintenance and repair are to be provided by the school, what is the average annual salary of the staff person needed for this purpose, including fringe benefits?

11. Can an adaptation of the tool to a more established technology allow much wider dissemination with only slight sacrifice of its central education features?

12. Can the new tool be used in courses as presently run? If it is successful, in what direction will it influence the development of these courses?

13. What publicity is required to encourage full use of the device?

14. How many students can use the services of this device simultaneously?

15. How many hours will each student use the device in the course of learning what it has to teach?

16. What is the cost per student hour, including overhead?

17. What is the *marginal* cost if one student uses the facility for one additional hour?

18. What is the additional capital cost, above present investment, to provide the service to a class of 10 students? 100 student? 1000 students?

19. If a school 100 miles away wants to install this system, what will be required in professional advice, in staff at the new location, in capital investment, in running costs per student hour, and in maintenance and repair?

ACKNOWLEDGMENTS

The author of this article organized and supervised the development of the spacetime display and accepts responsibility for its failure. The collaborators listed here have graciously allowed their participation to be acknowledged.

James Anderson of Stevens Institute; Dieter Brill of Yale University; Robert Brehme of Wake Forest University; Robert Cralle of Lawrence Radiation Laboratory, Livermore; Walter Daniels of IBM; David Finkelstein of Yeshiva University; Richard Lindquist of Wesleyan University; Arthur Luehrmann of Dartmouth College; George Michael of Lawrence Radiation Laboratory, Livermore; Wolfgang Rindler of the University of Texas at Dallas; Judah Schwartz of the MIT Education Research Center; Noah Sherman of the University of Michigan.

Simulated Visual Appearance of Rapidly Moving Objects

RONALD E. STONER

INTRODUCTION

One of the difficulties in using the simulation mode in undergraduate teaching is that, in order to be effective, it requires direct and immediate interaction between the student and the machine. The best results are achieved if a simulated experiment is treated like any other experiment, performed in the same environment and with the instructor at hand. These requirements eliminate the batch-mode processing of a simulation program. The student must either use a time-sharing terminal of some sort or have hands-on access to a computer that is ready for use at the scheduled laboratory time. Both of these alternatives would seem to make the simulation mode very difficult for schools with limited resources. Medium-size computers are still too expensive for the modest departmental budget, as are the monthly charges for commercial teleprocessing terminals. Even if a school has a larger computer capable of time sharing, there is still no guarantee that terminals will be located in the undergraduate physics laboratory and available for use during laboratory hours. So there are good reasons for examining what can be done in something approaching a simulation mode using equipment within the means of a smaller physics department.

A simulated experiment that requires only a machine capable of storing a few randomly accessible numbers and with limited programming ability is described in this paper. Several manufacturers have marketed moderately priced electronic calculators with at least this capability. The particular model we have used for three years in an undergraduate physics laboratory is a Wang Model 360, capable of storing four randomly accessible numbers. It can be programmed through a simple machine language in which each command represents a single keyboard button. It has no logic or branching capability, so it is a high-speed calculator rather than a computer. Even with these

Department of Physics, Bowling Green State University, Bowling Green, Ohio.

limited capabilities we have used it to do several so-called "simulated experiments" in the sophomore laboratory.

The particular experiment I will describe began as a recreation for me after I had seen a computer-animated movie[1] by Schwartz and Taylor which shows in perspective a simulated view of telephone poles along a road as viewed from an automobile speeding down the road at velocities close to the velocity of light. The textbooks I had read on the subject had always cited articles [2,3] which point out that, contrary to previous pedagogy, the effect of rapid motion on the visual appearance of the shape of objects was to cause an apparent rotation without change in shape instead of a Lorentz contraction in the direction of motion. Yet the movie showed that the poles, in addition to being rotated, were twisted and bent through large angles, apparently toward the center of the roadbed. I was certain that Schwartz and Taylor would not make a mistake so glaring, so I began to examine the origins of the apparent bending and twisting for myself.

As will be shown below, the result was a mathematically simple but computationally tedious formula for computing the visual appearance of a rapidly moving object. However, the calculator mentioned above had just been purchased and seemed ideally suited for the problem. The result of using the program to simulate a few objects not only explained to me what was going on in the movie but exhibited several other interesting effects, so the program seemed worth using as a teaching aid, and the natural place to use it was the sophomore laboratory.

THEORY

The basis for the experiment is usually introduced to the students by asking them to recall their experience with fast-moving, noisy aircraft. An airplane moving at speeds less than but near the speed of sound is usually not where an observer first looks for it in the sky. Instead it is much further along its trajectory than the direction the sound comes from would indicate. This is because the sound does not reach the observer's ear until some time after it leaves the source, and the airplane changes its position during that time interval. Similarly, if a source of light is moving at speeds less than but near the speed of light, it will be further along its trajectory than an observer would infer from the apparent direction of the light source.

To be more exact, let us suppose that an observer is located at the origin of his frame of reference and that some signal is emitted from a point source with the instantaneous coordinates (x,y). The z-coordinate can be taken to be zero by choosing the xy plane to include the straight-line trajectory of the source. Suppose also that the source is moving with speed v in the positive

x-direction. If the signal is emitted at time t and if the signal travels with speed c, the signal will arrive at the observer's eye at the later time T given by

$$T = t + (x^2 + y^2)^{1/2}/c \qquad (1)$$

At time T the source will have moved a distance $v(T-t)$ further along its trajectory from the point where the signal was emitted, so the source is not where the arrival direction of the signal would make it appear to be.

Now consider several point sources as making up the "shape" of a moving object. If the size of the object is significant compared with its distance from the observer and if its velocity is significant compared with the velocity of the signal, there will also be apparent distortions in shape. In that case the signals arriving at time T will have been emitted at several, possibly quite different, times t so that the relative positions of the points where the various signals were emitted may be quite different from the relative positions of the various sources at a given time. Both the apparent shape and the apparent position will be different from the actual shape and position.

When the signal is electromagnetic radiation, there are non-negligible differences between the shape of the object in its rest frame and its shape in the observer's frame of reference. To account for this additional distortion (i.e., Lorentz contraction), let us consider that the shape of the object in its rest frame is given by the set of constant positions of the light sources making it up. The positions of one of these sources has the coordinates (x',y'). If this frame is chosen so that its origin coincides with the origin of the observer's frame at time $t = 0$, and if the two frames are parallel, then the trajectory of a source is given in terms of its position in its rest frame by the equations of the Lorentz transformation between the two frames, which are

$$x' = \gamma (x - vt) \text{ and } y' = y \qquad (2)$$

To find the apparent position (x,y) of the point source with rest frame coordinates (x',y'), we need to solve Equations (1) and (2) simultaneously, eliminating t, to find expressions for x and y in terms of x', y', T, v and c. This is most easily accomplished by first eliminating x and solving for t, then using Equation (2) to find x. The result is

$$x = \gamma \left\{ x' + \beta\gamma cT - \beta \left[y'^2 + (x' + \beta\gamma cT)^2 \right]^{1/2} \right\} \text{ and } y = y' \qquad (3)$$

where β and γ are the usual constants v/c and $[1 - (v/c)^2]^{1/2}$, respectively.

Equation (3) allows one to calculate the apparent position of every point on an object according to the light that arrives at the eye of the observer at time T, so that it can be used to predict the apparent shape of the moving object. The chore of evaluating the expression by hand, even with the use of a desk calculator, is so tedious that the exercise would probably not be worth it. With a programmed calculator, however, the apparent position of each point can be determined in a few seconds with little effort. One can use the calculator to simulate an experiment that probably could not be done otherwise—that is, looking at some object that is traveling at a velocity close to the speed of light. The situation is sufficiently unfamiliar that the student does not know beforehand what to expect. In this sense, it is more of an experiment than most traditional ones.

RESULTS OF SOME SIMULATED EXPERIMENTS

Simulated experiments using the calculator to evaluate the expression for x in Equation (3), have been tried by students for three successive years in the laboratory of an honors section of a beginning physics course. The students have previously been introduced to special relativity in class and they have used the calculator before, so the explanation given above is usually sufficient instruction. They are asked to plan and conduct their own experiment by varying the parameters they have at their disposal—the size and shape of the object, its velocity, its trajectory, and the times T at which they wish to simulate its appearance.

The experiment usually starts with a student's sketch of a two-dimensional object in its rest frame on a piece of graph paper. The coordinates of several points are read directly from the graph and used as input to the calculator. The values of β, γ and T are stored in random-access registers and do not need to be re-entered before every calculation of the apparent position of a new point. The apparent position, newly calculated, can be immediately plotted by hand on a second sheet of graph paper, yielding a complete "picture" of the object in a few minutes. The plotted points are the locus of points from which the light was emitted that arrives at the observer's eye at a given time, so we call it "the apparent shape" of the moving object.

The student has the advantage of being able to look at this locus of points from many points of view, and he can thereby easily determine its true shape. The observer, from his single point of view, would have a much more limited set of visual clues. In the present two-dimensional examples the observer would see a line segment. In three dimensions (as in the movie[1]) he could only see the projection (two-dimensional) on his viewing plane.

Figure 1 is a sample of what a student might find although it is really the result of the first set of simulated experiments done with the program. It shows the apparent shapes and positions of an object which, in its rest frame, is a semicircle with its diameter lying along the x-axis, shown at various times as the semicircle passes the observer at a speed $v = 0.9c$. This particular illustration is used as an example to students of a simulated experiment to give them

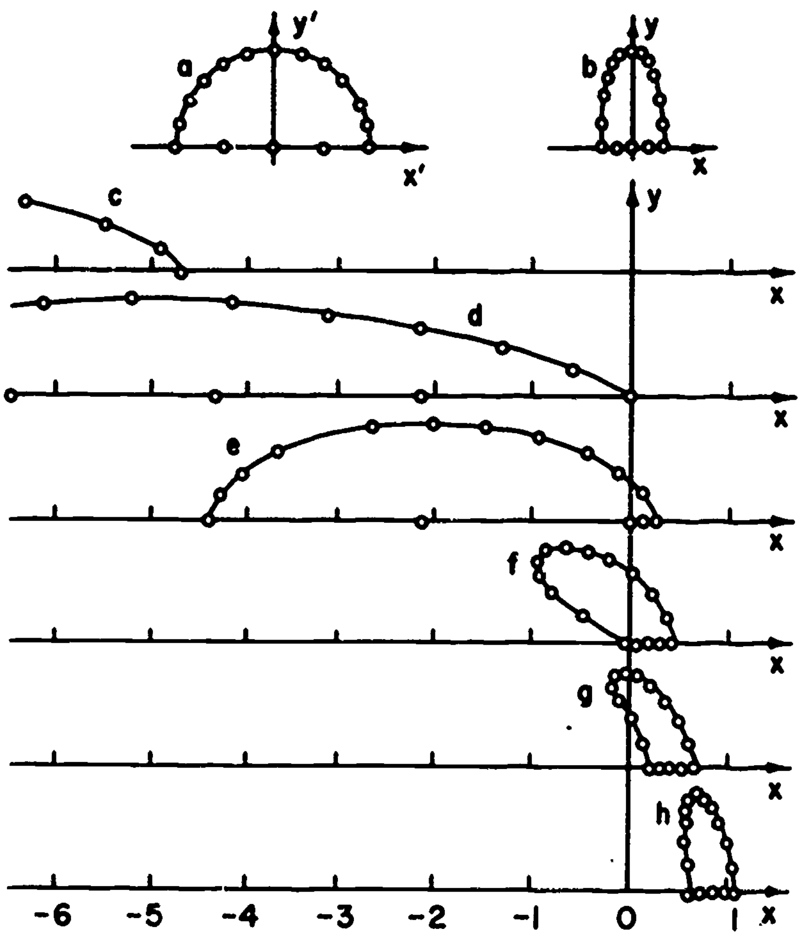


FIGURE 1 Changes in the visual appearance of a semicircle as it passes an observer at velocity $0.9c$. Radius = 1 light-second.

(a) The semicircle in its rest frame.

(b) The actual shape in the observer's frame is a Lorentz-contracted ellipse.

(c) At $T = -1.0$ seconds the semicircle approaches at an apparent speed of $0.9c$.

(d) At $T = -1/\beta\gamma = -0.4845$ sec the leading edge passes the observer at the origin.

(e) At $T = 0$ the center passes the observer.

(f) At $T = 1/\beta\gamma = 0.4845$ sec the trailing edge passes the observer.

(g) Even at $T = 0.1$ sec portions of the semicircle appear not to have passed the observer.

(h) At $T = +2$ sec the semicircle moves away at an apparent speed of $0.474c$.

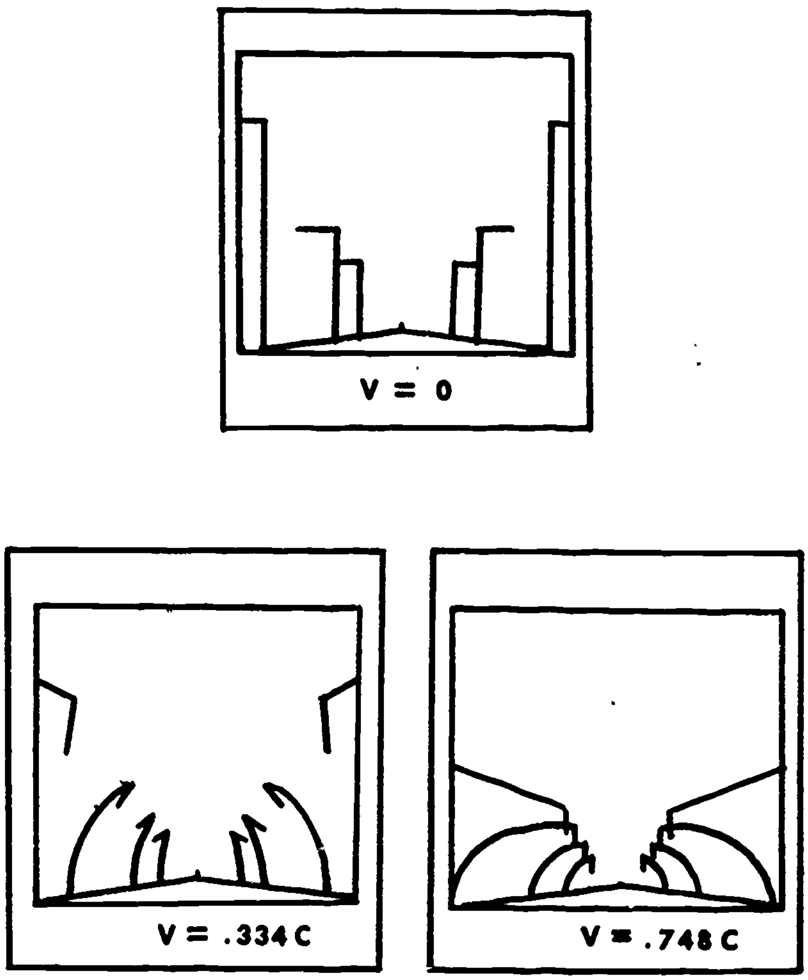


FIGURE 2 Replicas of three frames of the Schwartz and Taylor movie.[1]

an idea of what can be done. Several interesting and even surprising features emerge from this simple set of simulations. Notice, for example, the very rapid changes in the apparent shape and the size as the diameter passes the observer. The apparent speed is also interesting—the object appears to approach at several times the speed of light, then apparently abruptly decelerate to less than its true velocity as it passes. Finally, notice that there is a time during which the whole diameter of the semicircle has passed the observer, but several points on the circumference still lag behind in the second quadrant.

The illustrations in Figure 1 are sufficient to understand the apparent twisting and bending of the telephone poles in the Schwartz and Taylor movie,[1] some frames of which are shown in Figure 2. The points on an object that are furthest from the observer, such as the tops of the poles and outer portions of the crosspieces in the movie, always appear to be further back along the trajectory than the nearest points. When viewed in perspective against a background in a line drawing, where the observer has no sense of depth, this apparent bending backwards of the tops of the poles is interpreted to be a bending toward the roadbed center. If the object subtends only a small solid angle at the observer, like the crosspiece at the top of a pole, for example, the illusion is that of a simple rotation with no change in size or shape.[2,3]

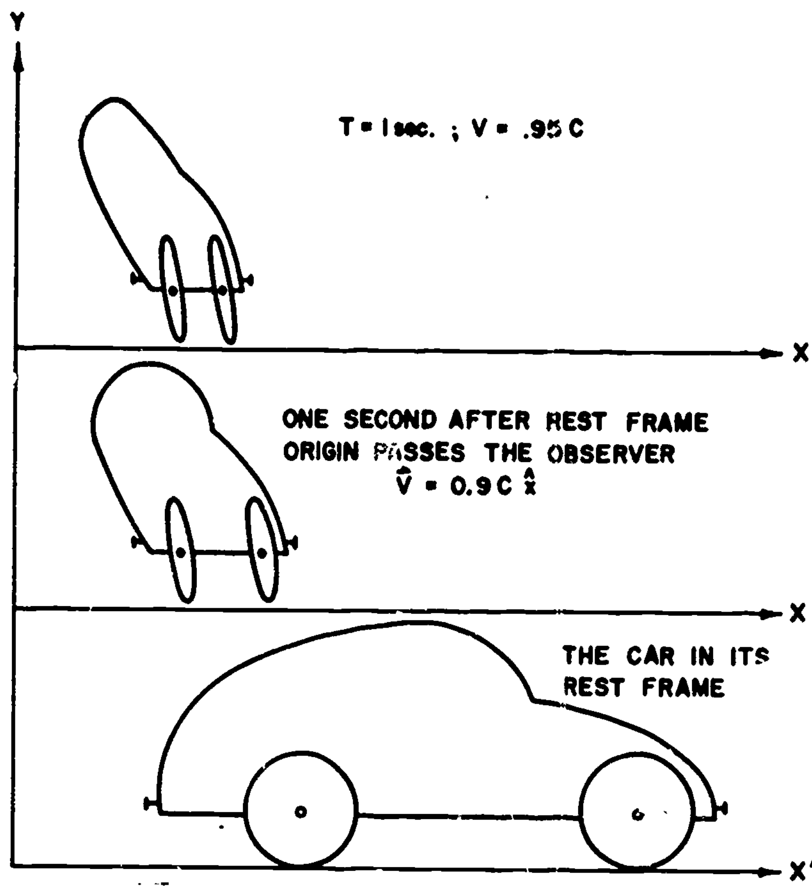


FIGURE 3 The visual appearance of a car after it has run over an observer at the origin at two different velocities. The car is 2.5 light-seconds long in its rest frame.

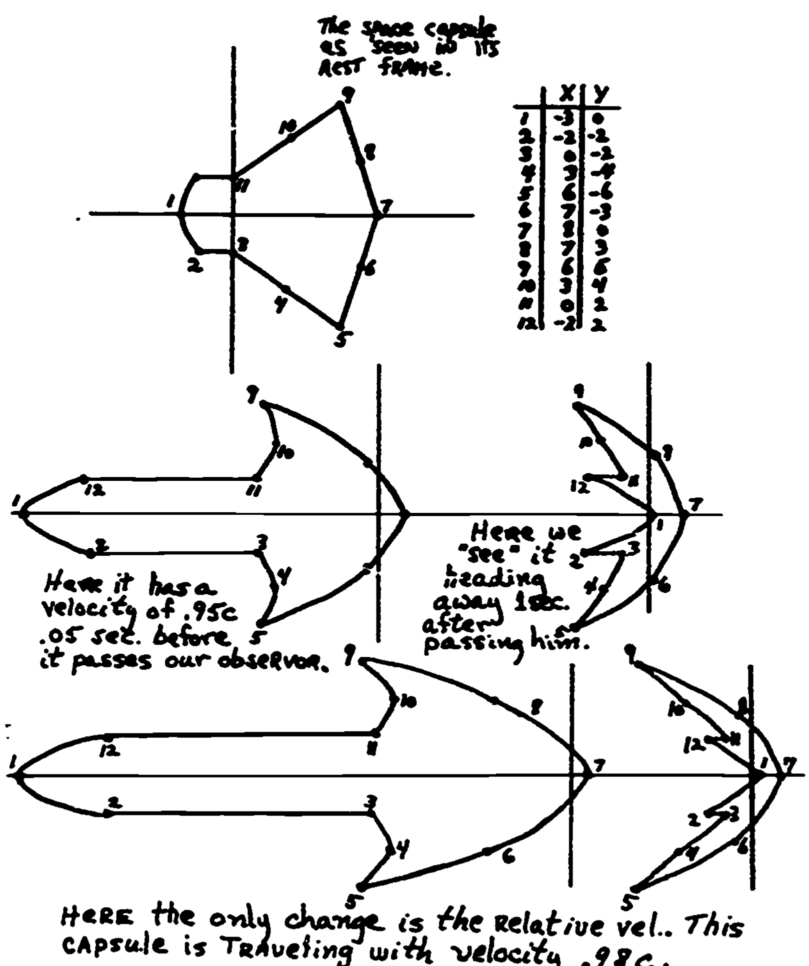


FIGURE 4 The visual appearance of a rapidly moving spacecraft, reproduced from a student laboratory report.

Students seem to enjoy dreaming up original and humorous objects to try, and are interested enough in seeing the results not to mind the hand plotting that is required. Figures 3 and 4 show some typical student-conceived and student-generated output. Figure 3 shows a car as it might look after running over the observer at three different velocities. Figure 4 shows two different velocities and two different viewing times for a space capsule. Other objects that have been simulated at various velocities and under various conditions include jack-o-lanterns, cups and saucers, stick men, rocket ships and the like.

The program is capable of a greater variety of simulations than are represented by the student-generated output shown. For example, it can be used to verify that the appearance of a small and distant object might seem to be undistorted from the rest-frame shape but rotated instead. A simple extension of the program will also allow the student to infer the color of the observed light from various points on the object, assuming the object emits monochromatic light in its rest frame. Both of these possibilities have not been tested with students yet, but a new crop arrives in the fall.

CONCLUSIONS

This particular experiment seems to have several characteristics that recommend it for teaching via the simulation mode. First, it does not seem possible to investigate the phenomena any other way except theoretically, by using approximations good in certain ranges of the parameters. Second, there are almost no traditional experiments in undergraduate laboratories that directly illustrate relativistic effects. Students are intrigued by ideas like Lorentz contraction and time dilation, so they approach the experiment with interest. Moreover, they like the ability to design their own experiment and they are usually surprised by the results.

Finally, the experiment can be done with a calculator that is multipurpose, yet relatively modest in price, so that even small physics departments can usually justify the purchase.

APPENDIX: INFORMATION FOR PROSPECTIVE USERS

Computer Used:	Wang Calculator Model 360.
Calculator Time:	Approximately 20 seconds calculation time per picture.
Language Used;	Simple machine language.
Extension to Other Machines:	Any digital device with .005K words or more, capable of taking square roots could be used.
Programming Effort:	Negligible.
Auxiliary Hardware:	None required.

Calculations on a bigger machine would probably be best on-line. A plotter or CRT output would be convenient but is not necessary. Copies of the Wang Calculator program and of laboratory hand-out sheets are available on request from the author.

REFERENCES

1. J. L. Schwartz and E. F. Taylor, "Computer Displays in the Teaching of Physics," *American Federation of Information Processing Societies Conference Proceedings 33,* 1285 (1968).

G. Schwartz, O. M. Kromhout, and S. Edwards, "Computers in Physics Instruction," *Phys. Today 22*, September 1969, p 41.
2. J. Terrel, *Phys. Rev. 116*, 1041 (1959).
3. V. F. Weisskopf, "The Visual Appearance of Rapidly Moving Objects," *Phys. Today 13*, September 1960, p. 24.

IV

ANALOG COMPUTING

Introduction

Since analog computers are relatively obscure and untried by most science teachers, we begin this section with a review paper by Martin which outlines the basic principles of analog computing. In the succeeding papers the formalism described by Martin is used by Burr and Brown to discuss problems in physics; Willett introduces the electrical engineer's viewpoint; and Liao, alone of the papers in this volume, undertakes to consider problems in ecology.

Analog computer capabilities have been upgraded in recent years through the use of terminals whereby several users are served simultaneously and by combining them with digital computers in so-called "hybrid" computers. In order to make their special characteristics more visible to the academic community, the analog computer papers were grouped into a separate session (Analog Computing). However, their educational applications may be expected to overlap those of digital computers.

Analog Computers in Science Education

DONALD C. MARTIN

INTRODUCTION

The word computer, to the layman, usually means a large, complicated machine that can perform numerical calculations or accounting functions at an extremely rapid rate. Today, everyone receives bills, letters, and bank statements produced by these machines, and it is natural that the word computer has become synonymous with digital computer and data processing. Unfortunately, this interpretation is shared by a large portion of the scientific community as well as the layman. One of the reasons for this is the widespread availability and versatility of the current generation of digital computers. With the advent of sophisticated, high-level programming languages such as FORTRAN and PL/1, digital computers have become very easy for the engineer or scientist to apply as a computational tool. A professional programmer is no longer required. There are literally hundreds of texts and programming aids to assist the novice in learning to use this tool. Almost all engineering and many physical science students graduate today with some competence in digital programming, and, in fact, the digital computer is being used in a great many high schools through time-shared remote terminals which use very simple programming languages such as BASIC or FOCAL.

There is, of course, another class of general-purpose computer, the analog computer. It would be foolish to argue that this type of computer is as versatile as the digital, and it is certainly not as extensively used throughout the scientific community. There are several reasons for this limited use, not the least of which are the limited availability of analog computers and certain misconceptions that have developed concerning the difficulty of their programming. In order to overcome this second limitation, the greatest need is for scientific and engineering graduates who understand what analog computers are, what types of problems are best suited to the analog computer, and how such problems should be formulated. Very few of our university

Department of Chemical Engineering, North Carolina State University, Raleigh, North Carolina.

faculty and graduating students today have any concept of the utility of analog computers.

It is important to understand that the analog is not just a cheap substitute for the digital computer. There are certain areas of research, design, and education in which the analog is simply the superior tool. In fact, recent developments have led to the marriage of the two machines as a *hybrid* computer for certain classes of problems. It is the purpose of this paper to introduce the operating characteristics and programming teachniques for the general-purpose analog computer and to introduce the concept of simulation with particular reference to science education.

Before proceeding with the operating and programming details, one further word might be in order as motivation for the application of analog computation in education. The imminent death of analog computation has been predicted by many individuals during the past ten years, but the field has survived these dire predictions. The recent growth of hybrid computation, along with efforts to couple experiments and processes with the digital computer, has given new impetus and direction to the analog field; very well expressed in an editorial by T. C. Gams.*

> "WE'VE SAID IT BEFORE, but it bears repeating—all but the submolecular phenomena in nature behave in an analog fashion, and the things we use to measure the parameters of nature *must* begin with an analog interface. Digitize as often and as early as you will: 'to this favor you must come'—an analog circuit must be involved.
>
> "MOREOVER, it is often wiser to digitize late in the measurement chain, after conditioning (scaling, compressing, filtering) the signal by efficient, economical, notably simpler analog means. Indeed, many analog techniques and tools have been so greatly improved, in recent years, that the A-to-D conversion line should probably be drawn much further to the right of the diagram than it would have been five years ago.
>
> "IN COMPUTATION, and in its sister sciences, Synthesis and Simulation, the distinctive merits of analog modelling remain unchallenged, despite all the gains of the digital technique (and of digital implements) in the past five years. For solving, simulating, or scaling problems with fewer than ten interdependent variables, no digital program will satisfy the needs of the scientist and engineer as fully and easily as will the corresponding analog approach.

*T. C. Gams, "On the Analog Art—Livelier than Ever," *Electronic Instrument Digest*, May 1970.

> "TALK TO the men who design instruments—analog, hybrid, or whatever kind—and you will probably find their attention focused on the analog interface. It holds the key to speed, range, linearity, accuracy, and stability. We would add resolution to that formidable list, but the analog circuit has infinite resolution . . . as infinite as the life expectancy of the analog art."

The point is, that an understanding of analog computer elements is of significant value to any scientist or engineer who is involved in the collecting, conditioning, and analysis of experimental data, whether or not he actually programs the computer. It is the intent of this paper to provide such an understanding along with the basic principles of analog computer programming and a brief introduction to digital simulation languages.

SIMULATION

Before introducing the programming concepts of analog computation, perhaps the word simulation should be defined explicitly within the context of this paper. Simulation has often been defined as the imitation of the actual behavior of some physical process or, in fact, of anything that can be defined as a system. In some cases it is convenient to simulate a system or process with an actual model such as the Link Trainer which uses an aircraft cockpit mockup to train pilots. Since the basic requirement for a simulation device is that its behavior is analogous to the original physical system, computers are often used as simulators when appropriate mathematical models for the system or process can be written. In many cases, such as the modern aircraft or space vehicle simulator, actual hardware is used in conjunction with computers to provide a truly realistic simulation of the interacting system composed of the flight vehicle and pilot.

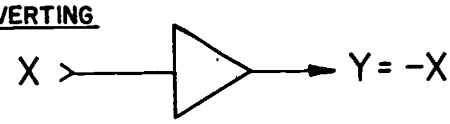
One of the advantages of the analog computer as a simulation tool is that the engineer or scientist retains a "feel" for the physical problem during the computation. If he wishes to determine the effect of increased cooling in a chemical reactor simulated on the analog computer, he can adjust a continuously variable parameter which is precisely analogous to turning a cooling water valve. The response of the simulated reactor can be immediately observed on a plotter or other display. The analog computer is solving the mathematical model of the real physical system, allowing the user to nondestructively test the system, evaluate the effect of various parameters on system operation, and observe the response of hypothetical system models. On the undergraduate level, there is the very real advantage of aiding the students' understanding of abstract mathematics. There is possibly no better way to impart an appreciation for ordinary differential equations than to employ the analog computer as a teaching aid, either in the laboratory or in classroom demonstrations.

ANALOG COMPUTER PROGRAMMING

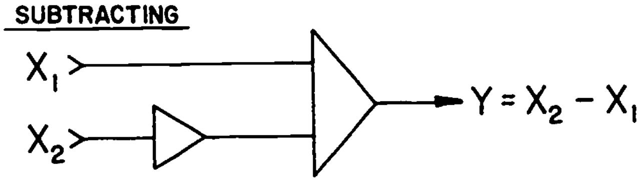
Analog computer programming is essentially a flow-charting process. There are certain conventional symbols that will be used in this paper to represent the mathematical operations that can be performed on an analog computer. These operations include summing, attenuation, multiplication, division, function generation, and integration. The reader interested in the hardware and detailed programming instructions will find several excellent sources given in the list of references.

The summer is an analog device for adding, subtracting, and inverting or changing the sign of variables. It is represented by a triangular-shaped symbol with one or more variable inputs and one output. Its output is the *negative* of the sum of all inputs.

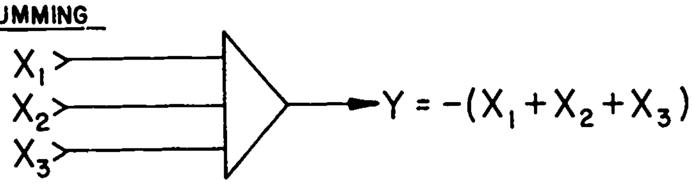
INVERTING



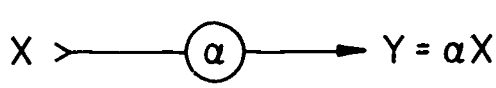
SUBTRACTING



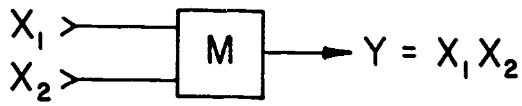
SUMMING

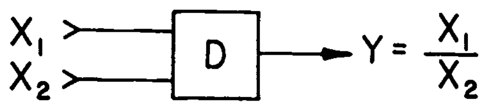


The attenuator or potentiometer is a device for multiplication of a variable by a constant factor less than unity. The programming symbol for an attenuator is

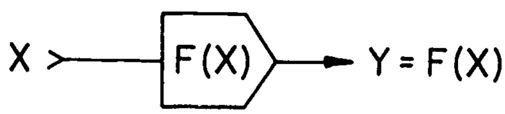


Two variables can be multiplied or divided on the analog computer. The symbolic representation of this device is

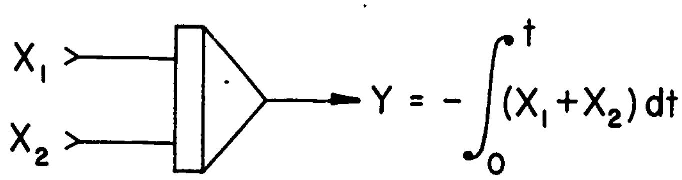




The generation of arbitrary functions of dependent variables is represented by this symbol:



The last operation, integration, is perhaps the most important element of the analog computer since it permits the continuous integration of sets of differential equations. The integrator can be used to sum variables as well as integrate and is represented by the following programming symbol. Like the summer, the integrator inverts the sign; i.e., the output is the negative integral of the sum of the inputs.

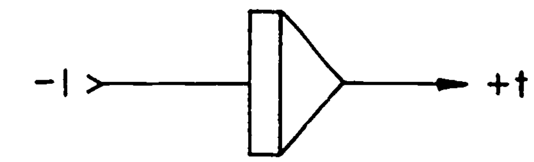


Programming an analog computer consists of interconnecting these computing elements in various ways in order to model physical systems. It should be noted that the independent variable in analog computation is time, i.e., all integrations are performed with respect to time.

The interconnection of these analog elements to obtain a desired function can be illustrated by programming the equation,

$$Y = a - bt + ct^2 - dt^3$$

Since the variable t is just a linearly varying quantity, it can be obtained by integrating a constant



The constant −1 represents the computer reference voltage. The integral of t is $t^2/2$, so we can integrate again to generate this variable, and it follows that a third integration will provide $t^3/3$.

The analog program for evaluating the above function $Y(t)$ is given in Figure 1.

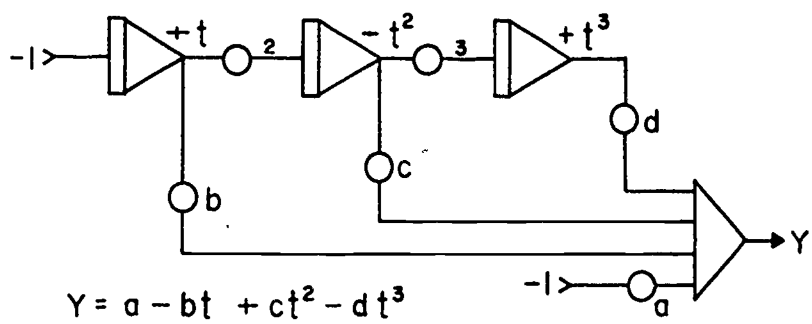


FIGURE 1 Analog computer diagram for polynomial equation.

You should not be disturbed if you do not understand the fine points of this computer program. The basic point here is that analog computer programming is very straightforward. Once you understand how the computing elements work, it is a relatively easy task to connect these elements to form the analog program.

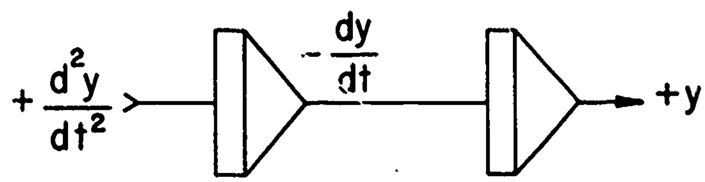
It is the capability of integration of a continuous variable which makes the analog so useful as a tool for the simulation of physical systems described by ordinary differential equations. To illustrate the technique, consider a spring-mass system at rest which might be described by the following equation

$$d^2y/dt^2 + (C/M)(dy/dt) + (K/M)y = (1/M)\,F(t)$$

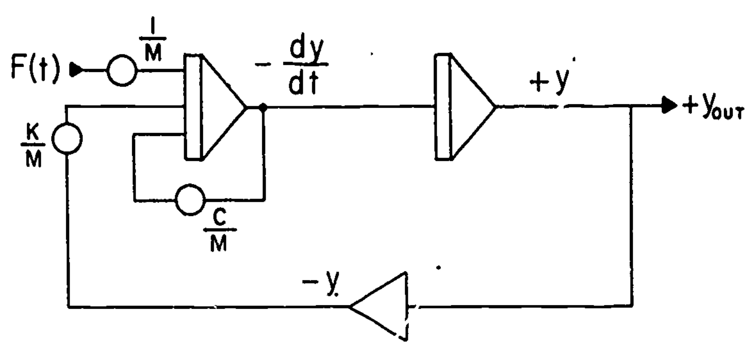
with initial conditions

$$dy/dt = 0, \quad y = 0, \quad \text{at } t = 0$$

This second-order equation would be solved for the highest-order derivative and integrated twice, i.e.,



It is then only necessary to multiply the derivative and dependent variable by the appropriate constants and provide the required forcing function to solve the given equation. The analog program would be as shown in Figure 2. The forcing function, *F(t)*, is supplied from other analog circuits or from an external function generator to obtain a solution.



$$\frac{d^2y}{dt^2}+\frac{C}{M}\frac{dy}{dt}+\frac{K}{M}y = \frac{1}{M}F(t)$$

FIGURE 2 **Analog computer diagram for a linear second-order differential equation.**

Of course, there is a little more to programming than shown in this simple illustration. Scale factors must be chosen to relate the real physical quantities velocity, displacement, and time to the analog computer variables, but the basic technique is certainly not difficult. The actual programming is accomplished by connecting the analog elements with wires on a patch panel. This patch panel is usually removable for program storage.

Perhaps a word about accuracy and the type of problem most suited to analog computation is in order here. Many digital computer proponents view the analog with disdain because of its limited accuracy. The accuracy of individual components is certainly limited by the precision with which the analog components such as resistors can be fabricated. Linear components such as summers and integrators can be purchased with accuracies on the order of 0.01%. Non-linear components such as multipliers have accuracies on the order of 0.5% to 0.02% of their full-scale voltage output. Since the combination of elements is dependent upon the problem, it is extremely difficult to make any general statement concerning the overall accuracy of the solution to any given program. The problem is further complicated by the fact that component errors are also frequency-dependent so that the total error is also a function of how fast the problem is solved. The cumulative experience of many analog computer users over the years leads to the conclusion that overall problem accuracy ranges from 0.1% to 5%. It is fair to state that the accuracy of the analog computer is limited to 1–2% for most problems, and that parameters and variables can be set or read to three or four significant figure at best.

With the class of problems most suited to the analog computer, this is not really a serious limitation. These problems usually consist of sets of ordinary differential equations which describe real physical systems. After all, parameters or physical variables can seldom be measured with much greater precision. Most data available in the literature are considerably worse. There is little point in using five significant figures for the heat transfer coefficients in a problem when the accuracy of the data used to obtain this parameter was ± 10 percent. In general, the analog computer is not suited to solving systems of algebraic equations that might describe the steady state of a system.

DIGITAL SIMULATION LANGUAGES

The ease of programming differential equations for analog computer solution has led to the development of digital simulation languages. These high-level languages implement analog computing elements in a discrete manner on the digital computer. They allow the user to program the digital computer directly from a block diagram or flow sheet very similar to an analog computer diagram.

As with any high-level language, simulation programs require translation from the input or source form to the binary machine language of a particular digital computer. This translation is performed by a digital simulation processor which can be either an interpreter or compiler. If the translation is completed prior to execution of the simulation program, it is called a compiler. The interpretative program executes statements of the simulation program as they are encountered. A few of the many simulation languages which are in use today include: continuous systems modeling program (CSMP), digital simulation language (DSL), continuous system simulation language (CSSL). Some simulation languages allow the insertion of user-written FORTRAN or ALGOL subroutines. If many simulation runs are anticipated by changing parameters within the program, the compiler type is inherently more efficient. Conversely, the interpreter requires more computer time for each run, but it offers the added flexibility for program or model changes after each run.

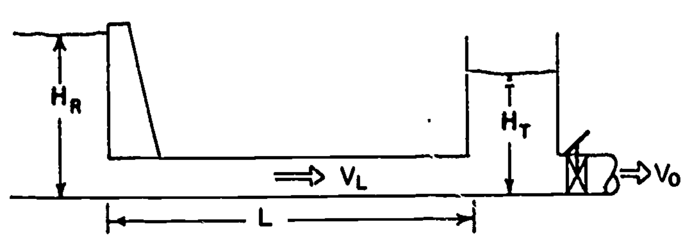
Digital simulation languages normally sort input statements before execution and allow the user to choose an appropriate integration routine and interval. The user's choice of step size and error criteria affects the accuracy of the simulation and the amount of digital computer time required. These languages generally provide graphic output on a printer, plotter or oscilloscope. (Many applications of digital simulation languages are given in Chu.[1])

To illustrate the use of a digital simulation language and emphasize the similarity to analog computer programming, consider the following problem[2]:

Suppose we wish to simulate the dynamic response of a surge tank and pipeline system as shown in Figure 3. The response of the level in the surge tank is desired when the outlet valve is suddenly changed from the fully open to the closed position. If the surge tank is open to the atmosphere, the mathematical model for the system is given as the simultaneous equations:

$$(L/g)\dot{V}_L = H_R - K_1 V_L \,|V_L| - H_T + K_2 \dot{H}_T \,|\dot{H}_T|$$

$$A_T \dot{H}_T = A_L (V_L - V_0)$$



H_R – RESERVOIR HEAD
A_L – LINE AREA
A_T – TANK AREA
K_1 – LINE LOSS COEFFICIENT
K_2 – TANK LOSS COEFFICIENT
L – LINE LENGTH
V_L – UPSTREAM LINE VELOCITY
V_0 – DOWNSTREAM LINE VELOCITY
H_T – FLUID LEVEL IN TANK

FIGURE 3 Surge tank and reservoir system for digital simulation illustration.

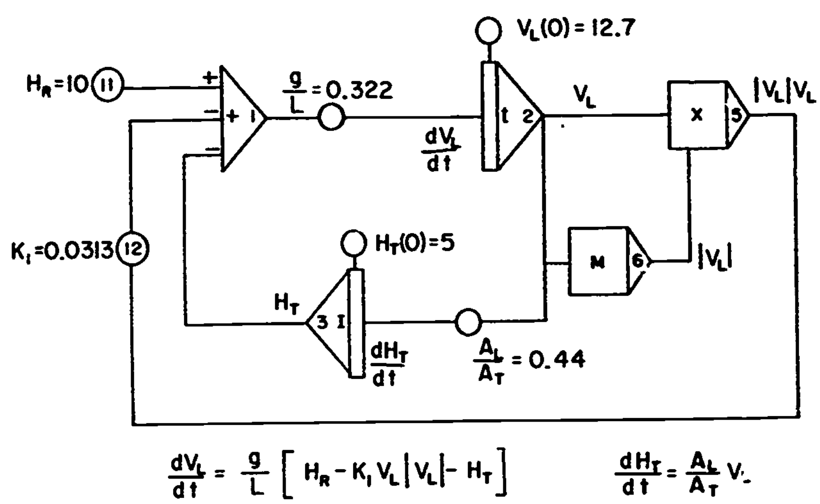


FIGURE 4 Digital simulation program for reservoir surge tank problem.

The digital simulation flow diagram for solving these equations is given in Figure 4. Note that each mathematical operation is essentially equivalent to the analog computing symbol for the continuous function desired. A listing of the input for a specific simulation language, CSMP, is as follows. It is assumed that K_2, the tank loss coefficient, is zero for this case.

CONFIGURATION SPECIFICATION

Output Name	*Block No.*	*Type*	*Input 1*	*Input 2*	*Input 3*
LINE VELOCITY	2	I	0	1	0
VELOCITY SQR	5	X	6	2	0
FLUID HEIGHT	3	I	0	2	0
SUM	1	+	11	–3	–12
RES HEAD	11	K	0	0	0
LINE LOSS COEF	12	G	5	0	0
ABS VALUE VEL	6	M	2	0	0

Each block is numbered and identified by type, i.e., integrator (I), multiplier (X), summer (+), etc. Up to three inputs may be specified for most blocks, and constants can be entered in a block labeled K as indicated for the reservoir head. Use of these configuration specification statements is equivalent to patching or wiring a program for the analog computer.

After the configuration statements tell the computer how functional blocks are interconnected, it is necessary to assign initial conditions and constants or parameter values. Again using CSMP to illustrate the technique, these values are supplied as follows:

INITIAL CONDITIONS AND PARAMETERS

IC/Par Name	*Block*	*IC/Par 1*	*Par 2*	*Par 3*
VEL IC G/L	2	12.7000	0.3220	0.0000
HT IC AL/AT	3	5.0000	0.4440	0.0000
LOSS COEFF	12	0.0313	0.0000	0.0000
RES HEAD	11	10.0000	0.0000	0.0000

This operation is equivalent to setting potentiometers on the analog computer. We see from this table that the initial value of the velocity is 12.7 ft/sec and the input coefficient for this integrator block is G/L = 0.3220. Reference to Figure 4 shows that each constant, initial condition, and parameter is specified by block number in this list. At this point, the computer program requests an integration interval, integration routine specification, and the outputs to be used for output display. A plotter output for this particular simulation is shown in Figure 5.

There are many advantages of digital as compared to analog simulation. If floating-point hardware is available, the equations are easy to scale. There is no patching required and very little worry about equipment failure. One major disadvantage of digital simulation is the lack of hands-on or interactive operation except in those instances where very sophisticated and expensive graphic output is available. Interactive graphic display equipment can easily cost from fifty thousand to one hundred thousand dollars. A second disadvantage is the speed of computation. The simple example discussed above can be solved repetitively by an analog computer in a time frame of from one to twenty milliseconds. Any changes in the system response can be instantaneously observed on an oscilloscope as parameters are varied continuously. The digital simulation of this system took approximately six minutes to provide the plotted output shown in Figure 5. For undergraduate student use, the analog has the real advantage of speed and interaction when they can control the simulation. On the other hand, digital simulation is easier for most instructors to learn and can be used for simulation and demonstration of much more complex physical systems.

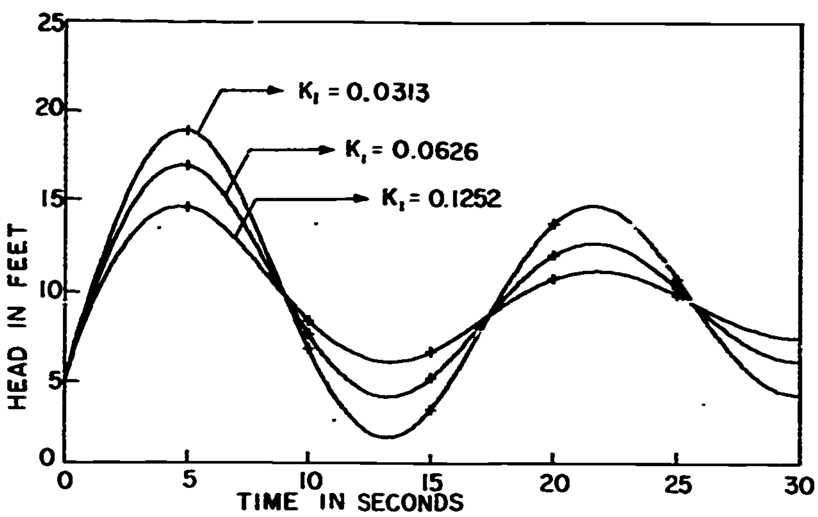


FIGURE 5 Plotter output for simulated surge tank.

EDUCATIONAL USE

All educators active in computing are well aware of the problems of computer access in a university atmosphere. In almost all cases there is a tendency to insist that university computers share a dual role in the areas of research and education. Unfortunately, these areas are not hardware compatible in the case of analog computers. The researcher wants one fairly large computer with a great deal of non-linear equipment, while the beginning students are better served by a number of smaller, less-sophisticated computers. Limited funds usually preclude the establishment of a laboratory to serve both areas adequately. Additional patch panels are not the answer. Consider for example a laboratory section of twelve students, all expected to use the computer during a two-hour laboratory period. The optimum number of students using the computer at any one time is about two or three. Even with four patch panels available, all groups will patch the problem in about the same time, and the instructor will have to chase the students away from the machine just as they are beginning to experiment with their solution. Thus, the most valuable time for the student is cut short and he retains forever the impression of the analog computer as a programming and patching exercise and not as a useful tool for simulation or an aid in the understanding of the dynamic response of physical systems. Present hardware limitations severely restrict in-class use of computers for large numbers of students at the same time. Digital simulation suffers from the same disadvantage because of a lack of inexpensive interactive graphic display terminals and the long solution time.

Since most undergraduates will never see or use an analog computer after they leave the university, we must question the value of teaching them patching details. This does not mean that they should not understand and use analog elements as signal conditioning devices or should not do the actual programming for simple first- and second-order systems. It does mean, however, that they should not have to expend their valuable time learning all the intricate details of a specific computer to program six-degrees-of-freedom problems or complete nuclear reactor simulations. For this type of problem, they should have access to a prepatched and debugged program so that they can study the physical system and not the computer. One way to accomplish this is through the use of analog computer terminals.

Analog computer terminals are a relatively new development and have only become commercially available within the past two years. These terminals allow any of several problems to be selected by each of sixteen or more

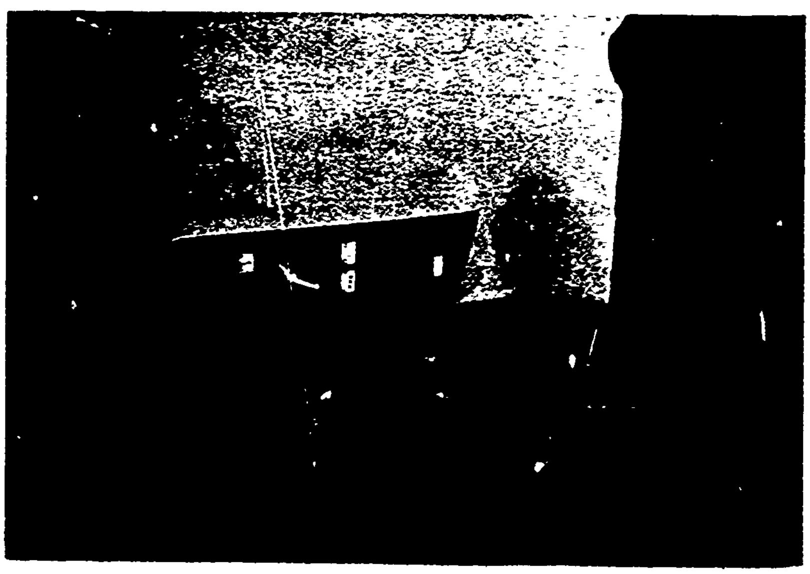


FIGURE 6 Analog computer terminal for student use.

students in a classroom. The student has control of the problem parameters and displays his output on a storage screen oscilloscope. He can work at his own speed but has a response time on the order of a few milliseconds if desired. He can plot and store several outputs on the oscilloscope screen, then change parameters to immediately observe their effect on the simulated system.

A simulation system using terminals such as shown in Figure 6 has recently been installed at NCSU.[3] The potential of these analog computer terminal systems has yet to be realized, but the key to their acceptance and use will undoubtedly lie in the development of appropriate software for various curricula. If such programmed instruction software becomes available with terminals on the order of $2000 each, we may see radical changes in our basic science and engineering laboratory instruction during the next few years.

REFERENCES

1. Y Chu, *Digital Simulation of Continuous Systems*, McGraw-Hill, New York, 1969.
2. D. J. Wood, "Lumped and Distributed Parameter Analog Analysis of Transient Liquid Pipe Flows," *Analog/Hybrid Computer Educational Users Group, Applications, IV, 3*, 21 (1969).
3. D. C. Martin, "Development of Analog/Hybrid Terminals for Teaching System Dynamics," *AFIPS Conference Proceedings for Fall Joint Computer Conference*, p. 241, 1970.
4. J. E. Stice and B. S. Swanson, *Electronic Analog Computer Primer*, Blaisdell, Boston, 1965.
5. J. J. Blum, *Introduction to Analog Computation*, Harcourt, Brace, and World, New York, 1969.
6. R. P. Jenness, *Analog Computation and Simulations: A Laboratory Approach*, Allyn and Bacon, Boston, 1965.
7. A. S. Jackson, *Analog Computation*, McGraw-Hill, New York, 1960.
8. T. D. Truitt and A. E. Rogers, *Basics of Analog Computers*, Rider, New York, 1960.

The Analog Computer in the Classroom

ALEX F. BURR

INTRODUCTION

This paper describes the use of an analog computer in the classroom to demonstrate the solutions of the Schrödinger equation for a finite square-well potential and, in particular, to find the eigenvalues of the problem. A description of the programming of the computer is given. Some comments on the utility of this type of demonstration and on its reception by the students are also given.

With a few notable exceptions the computer has not yet had much impact on the activities inside the physics classroom. One of the reasons for this state of affairs is that the individual instructor does not have time to reorganize his whole course in terms of the computer nor is his school able to afford an elaborate computing center for classroom use. Therefore, the busy instructor does just what he did last year—namely, lecture and draw some diagrams on the chalkboard.

If the computer is to have a significant influence in many classrooms, the instructor must not have to reconstruct his whole course or spend much additional time in preparation. In physics courses, much time is spent in studying the mathematical relationships between physical quantities, usually expressed in terms of differential equations. Hence it was decided to computerize the presentation of one differential equation, the Schrödinger equation, in such a way that the instructor could insert this unit into his course whenever he wished.

It was soon realized that an analog computer was more suited for the solution of differential equations than a digital computer. For differential equations it is much faster. Furthermore, its output, which is usually in the form of a graph, is much more useful in a class than the output of a digital computer, which is usually in the form of a table of numbers. The analog computer possesses for classroom use a combination of speed and economy which no other computing system can approach.

The analog computer does have some disadvantages. It is not so accurate, nor so flexible, as a digital computer. Furthermore, it is harder to program

Department of Physics, New Mexico State University, Las Cruces, New Mexico.

logical decisions on an analog computer. However, none of these features is important in this application. The major programming difficulty of analog computers is the proper scaling of the problem. However, when one wants to demonstrate the character of a differential equation rather than to solve explicitly one example, the scaling problem becomes irrelevant.

THE PROBLEM

The simple one-dimensional Schrödinger equation for a wavefunction $\psi(x)$ with a square-well potential

$$d^2\psi/dx^2 + (2m/\hbar^2)\,[E - V(x)]\psi = 0$$

$$V(x) = \begin{cases} 0, & -a \leqslant x \leqslant a \\ V_0 > 0, & |x| > a \end{cases} \qquad (1)$$

was selected for this demonstration because (a) it represents a non-trivial application of the analog computer, (b) it describes an exceedingly important physical relationship, and (c) the parametric nature of the total energy E represents an idea at once difficult for the student to grasp and ideally suited for the analog computer to demonstrate. $V(x)$ in this case is a square-well potential of the type shown in Figure 1. This equation can be programmed for an analog computer if it is realized that x can be treated as the time variable and that the symmetry of the problem permits one to use only the right half of the potential well. Thus one can start the even parity solutions with an arbitrary $\psi(0)$ and $\dot{\psi}(0) = 0$ and the odd parity solutions with $\psi(0) = 0$ and arbitrary $\dot{\psi}(0)$.

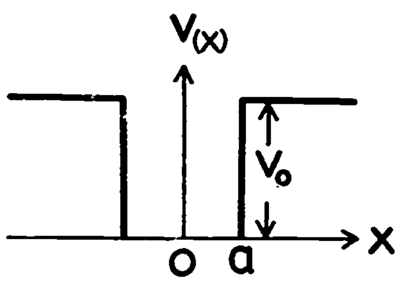


FIGURE 1 Square-well potential to be used with the Schrödinger equation.

The symmetrical nature of the problem is requisite for its solution in this manner. The analog computer necessarily works in the time domain; hence, for a second-order differential equation, it must be provided with two initial conditions. A reasonably simple analog computer cannot handle boundary conditions directly. In this problem, the arbitrary value of the non-zero initial condition only affects the amplitude of the solution, not its shape. Thus, if one were trying for an exact solution, one would use the normalization condition to set its value, but for the purposes of demonstrating the nature of the solutions one really doesn't care whether the area under the curve $|\psi|^2$ is unity or not.

Although the wavefunctions are composed of a combination of simple harmonic and exponential functions, a transcendental equation must be solved if the eigenvalue, E, is to be found directly. However, for any arbitrarily chosen value of E, the analog computer will quickly give the appropriate solution for the wave function. Those solutions that are quantum mechanically well-behaved can then be identified and the corresponding eigenvalues of E recorded. The way in which the computer permits one to vary E clearly demonstrates to the student the reason why only certain values of E are allowed on physical grounds. For ease in understanding the programming of the computer, equation (1) should be written in the form

$$\ddot{\psi} = KV(x)\psi - KE\psi \qquad (2)$$

where K is the constant that results from combining $\hbar$ and m.

THE COMPUTATION

The analog computer circuit diagram is given in Figure 2. In this diagram the circles represent potentiometers and the other shapes specialized circuits based on operational amplifiers. Each component has been numbered for ease of reference, and many of the connecting lines have been labeled with the part of equation (2) which the voltage on the line represents. The circuit is drawn for an EAI model TR 48 computer but can be readily adapted for other computers.

That part of the diagram which is enclosed in dotted lines represents a function generator whose purpose is to create the particular potential function desired for the problem at hand. In this case it is set up to generate the step-function that represents the potential of Figure 1 for positive x. The potentiometer 5 and integrator 7 generate a ramp voltage which causes the comparator 23 to change states when the ramp voltage equals the voltage from potentiometer 13. When the comparator changes its state, it causes the output of switch 22 to go from zero to a value selected by potentiometer 12. This value is V_0. Potentiometer 13 controls the time that the change in V takes place; hence, sets the value of a.

Component 11 is a multiplier which combines $V(x)$, the output of switch 22, with $\psi(x)$, the output of integrator 21. The inverters 9 and 10 are present because the multiplier needs positive and negative versions of the quantities to be multiplied. In this particular case where the function generated is a step

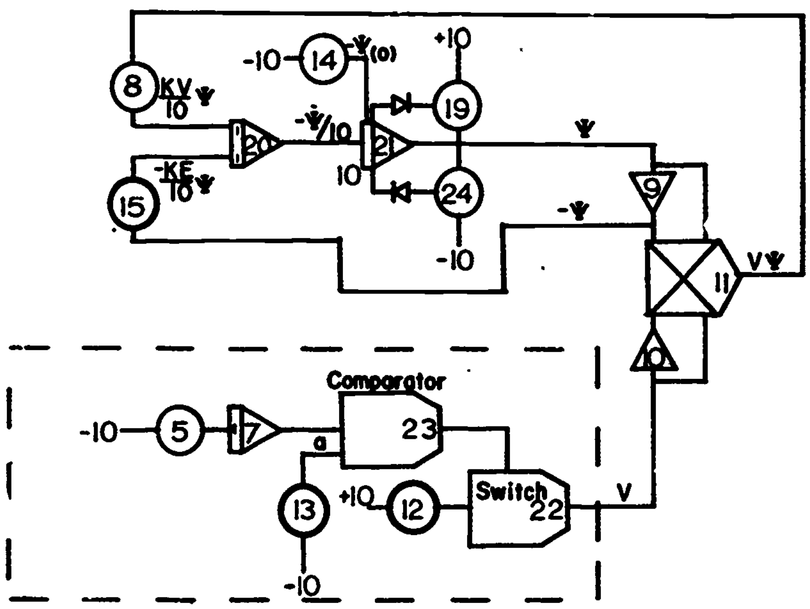


FIGURE 2 Analog computer circuit diagram.

function, the multiplier is not really needed.* However, if a more complicated potential such as the harmonic oscillator potential is desired, a multiplier must be used. The product of ψ and V is then fed through potentiometer 8, which can also be used to set the size of the potential step, to the input of integrator 20. The other input to integrator 20 is $\psi(x)$, which is obtained from inverter 9 through potentiometer 15, which sets the value of E. The sum of these two quantities is the second derivative of $\psi(x)$; hence, the output of the integrator is the first derivative of $\psi(x)$ which is fed to integrator 21. The signal is fed to the X10 input in order to keep the voltage output requirements on integrator 20 below the overload level. The output of integrator 21 is $\psi(x)$, the solution to the equation. Potentiometers 19 and 24 and their associated diodes are used to prevent the overloading of integrator 21 which would otherwise occur if ψ for values of x greater than a were allowed to go toward $\pm\infty$ as it ordinarily would (except for values of E equal to the eigenvalues). The output of switch 22, $V(x)$, and the output of integrator 21, $\psi(x)$, are simultaneously displayed on an oscilloscope to show the solution of the equation superimposed on the form of the square-well.

If the computer is programmed as shown, the solutions displayed are the

*D.L. Shirer, *Analog/Hybrid Computer Educational Users Group Applications 2*, No. 3 (1970), describes a circuit that does not use the multiplier and gives directions for wiring up a simple comparator.

even parity solutions, because the initial conditions (zero for integrator 20 and an arbitrary value set by potentiometer 14 for integrator 21) correspond to $\dot{\psi}(0) = 0$ and $\psi(0)$ = amplitude of ψ at $x = 0$, respectively. If the odd parity solutions are desired, transfer the output of potentiometer 14 from integrator 21 to integrator 20.

The exact setting of the various potentiometers is unimportant for demonstration purposes. The following set of values can be used to set up the problem initially:

Potentiometer	5	8	12	13	14	15	19	24
Setting	0.04	0.1	1.0	0.2	0.4	0.007	0.5	0.5

The settings are such that the step in the potential is about as big as practical, the step takes place about halfway through the repetition cycle, and the energy is set for the first even eigen-solution. There is, for this particular well, one other even eigen-solution which appears when potentiometer 15 is set to 0.058. The computer is set in the repetitive operation mode so that the effect of a variation in any of the parameters on the solution can be immediately noticed.

The above explanation of the analog circuit has been simplified by neglecting some details such as the sign inversion which accompanies the passage of the signal through each amplifier and the scaling technique which was used to keep integrator 20 from overloading. These details can be neglected because, once the computer is programmed, all that is needed to demonstrate the solutions to the finite square-well problem is to remember that potentiometer 12 sets the depth of the well, potentiometer 13 sets the width of the well, and potentiometer 15 sets the value of E.

It is well known that the solutions will in general rapidly tend toward infinity for values of x greater than a. However, for certain special values of E, the solutions for large x will approach zero. It is these solutions that have physical meaning, and it is these values of E that one spends his time searching for. Without a computer, the task is very tedious and time-consuming. With the computer, all one has to do is turn one knob and watch the solutions approach zero for large x. Figure 3 attempts to convey some of the feeling for the problem that one gets while watching the solutions change as E is slowly brought up to and past the eigenvalues. It is hard to describe in words the excitement one feels when the exponential tail of the wavefunction is made to flatten out as the value of the E potentiometer approaches the eigenvalue.

RESULTS AND CONCLUSIONS

This presentation has been well received in the classroom. Students have

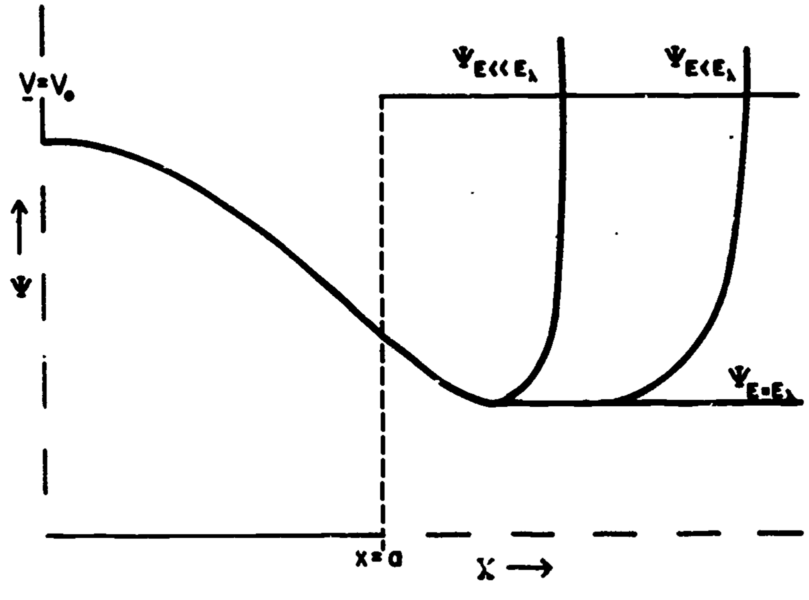


FIGURE 3 Several solutions to equation (1) as E is made to approach eigenvalue.

indicated that it helped them to visualize the square-well solutions and the behavior of ψ outside the well. It has been a particular aid in understanding the reason for the quantization of the square-well energies. Some students were disturbed by the fact that only the right half of the problem was shown, and others had difficulty in accepting the analog computer as a blackbox which produced solutions and felt that they had to know more details about how it produced the solutions before they could comfortably accept the answers.

The analog computer presentation worked very well for small groups of students. Large classes, however, presented additional problems. The main problem was that the computer was not located near a classroom and was difficult to transport; furthermore, the usual oscilloscope on which the solution is presented is too small for a large class to see comfortably. This problem was overcome by having a simple "home movie" made of the computer oscilloscope as the problem was run.

This procedure does have the disadvantage that the solutions are not before the class at all times. Also there is some problem in getting the oscilloscope traces to appear clearly on the film. However, once the film is made (and it is surprisingly easy to make) the demonstration is available for use in later classes with no additional effort on the part of the instructor. Of course, the demonstration is not so flexible as it would be if given live.

Another solution to the physical problems of computer location and large classes is closed-circuit television. A camera located at the computer with a monitor in the classroom has been successfully used and does meet the above objection to a movie; however, the minimum equipment cost is about $500. When this solution was first tried there was some question as to whether the TV camera would miss some of the oscilloscope traces. These fears proved groundless.

In the future it is expected that this technique of computerizing the presentation of physics equations will be tried on various combinations of computer hardware, with various types of equations, and at varied levels in the physics curriculum. The end result should be a tested set of presentations requiring a minimum of effort on the part of the instructor which he can insert into his course with a minimum of change in his usual course. The student should gain in understanding because a clearer presentation of the physical relationship involved in the given equation is possible since the mathematical manipulations involved will be largely performed by the computer.

Don Merrill and Philip Dillard from New Mexico State University and Robert Owens of the Air Force Academy provided particularly helpful assistance with the project upon which this article is based.

Analog Simulation in an Experimental Setting

ROBERT M. BROWN

INTRODUCTION

This paper discusses the meaning of simulation as implemented in an experimental setting, some features of the hardware and software needed for such simulations together with the advantages of an analog or analog/digital hybrid approach, and the ease with which one can "do it yourself," as illustrated with a detailed example, the Quadrupole Mass Spectrometer (QMS). The discussion does not assume a familiarity with the general topic of simulation or with analog computation but is intended to encourage teachers to consider such an approach and to underline how readily they can begin to use this technique in their teaching. Some definite suggestions will be made in this regard.

Some of the ideas presented below were explored and checked out during the past two years in a project[1] entitled "Computer Simulated Problems and Experiments in Physics" (CSPEP). This project was concerned with the design, construction, and checkout of a prototype Graphics Display and Control Console (GDCC) and a wide range of simulations for science and engineering teaching. It originated in an attempt to develop a facility that would allow the student to concentrate on the physical phenomena described by a model or theory, the *physical* meaning of the parameters of the model, and the *changes* of these parameters. Analog and hybrid techniques allow an extremely fast interactive computer graphics approach in science education. In order to allow the student easily to control and use such a computer facility, the GDCC was built to resemble an experiment monitoring station used in professional research activities, but it is activated by a preprogrammed simulation model. Such a control console provides the student (and the scientist) with maximum graphical and visual display of information

---

Field Services Division, Computer Sciences Corporation, Huntsville, Alabama. This work was begun in the Department of Physics, University of Alabama, Huntsville, and supported in part by a Loan Contract with the U.S. Army Missile Command and by ASEE-NASA Faculty Fellowships at Marshall Space Flight Center, Huntsville, Alabama.

FIGURE 1 A student using the Graphics Display and Control Console (GDCC) in the pilot course.

and data and extreme flexibility of control of phenomena. A photograph of the early version of the GDCC appears in Figure 1. As one can see, it contains strip chart recorders, meters, helipotentiometers, switches, CRT display, counters, and signal generators. An *XY* recorder is just to the left of the picture. Future versions will also contain a digital keyboard, teletype, and analog pen. This combination of equipment is similar to a general data acquisition and control system and can indeed function as such in an instrumented laboratory approach.

IMPLEMENTATION IN AN EXPERIMENTAL SETTING

The term Simulation in an Experimental Setting (SES) is intended to combine two concepts: (1) computer modeling and investigation and (2) real-time, hardware-oriented student/computer interaction. In science teaching it is important to help the student to develop an understanding of the concept of models (theories), to gain familiarity with the currently accepted ones, and to be able to create models of his own. Often he works only with simplified equations that are amenable to closed-form mathematical analysis.

These models usually lack detail and sufficient complexity to be reasonably good representations of the real world, and the above goals are not really achieved but hidden. However, with the use of computers the desired goals should be attainable.

Students can certainly investigate models in a more meaningful way through a batch processing mode with a large digital computer. However, two things are missing in such computer modes—good graphical display and student interaction and control of the model. It seems more appropriate to use a name other than *simulation* for such non-interactive modes, e.g., "mathematical model analysis." With graphics-oriented, *real-time* interaction the student can really use the computer to study, control, and investigate a wide range of significant models and to appreciate their physical importance. The term real-time simulation means that the phenomena considered have a time development proportional to the motion of the hands of a real clock on the wall. This is usually implemented in a digital system by an external timer which holds the beginning of each successive computation loop until a given number of milliseconds has elapsed and the external timer/clock gives an interrupt signal to the computer. In analog computation the "real-time clock" is inherent in the way the integrations are performed (charging of capacitors). In hybrid work both a real clock external to the digital system and the "built-in" analog clock must be synchronized.

There are several pedagogical hypotheses about the nature of learning that are implicit in what has been said and in the discussions that follow:

(1) The student should be *active* in his learning experience and should be able to exercise control over his learning environment.
(2) Learning should be goal-oriented with structured steps where appropriate.
(3) Learning requires decision-making and involves practice.
(4) The student should witness promptly and without penalty the results of his own decisions.
(5) He should implement his judgments, if possible, in terms of physical hardware in order to help him obtain a concrete meaning of the physical/mathematical concepts he is studying.
(6) Visual clarity of concepts is very important and eye/hand correlations should be used where appropriate.
(7) The learning environment should provide fast psychological feedback and reinforcement with motivation based on success, not punishment for failure.
(8) The student should experience only legitimate frustration in-

volved in science not artificially imposed difficulties often found in student laboratory experiences.

(9) The student should control a simulated problem/experiment in terms of the concepts he is investigating not in terms of computer hardware and software.

(10) The learning process should be self-paced with an opportunity for review at will.

(11) The student should investigate a wide range of significant experiment/problems and use such experience as a *basic* and *necessary* means of developing his understanding.

With this framework in mind it is appropriate to present a concrete example to illustrate the above ideas and indicate the meaning of SES in a specific problem.

This problem/experiment was developed and used in an actual course during the CSPEP Project.[1] Consider that we have a class of Introductory Physics students using one of the standard calculus-based texts (Halliday and Resnick[2] was used in the CSPEP course) and studying the relationship of magnetic fields to current sources. The fundamental relationship with which we would like the student to gain familiarity is the Biot-Savart Law:

$$\mathbf{dB} = \mu i\, d\boldsymbol{\ell} \times \mathbf{r} / 4\pi r^3$$

One should make clear that it is not the purpose of the following simulation (or any other simulation for that matter) to convince a student of the validity of this or any other physical law. Rather, it is to provide him with an environment in which he can gain experience with the physical meaning of the law or model. In the case of the Biot-Savart Law, the simulation will allow him to "see" its meaning in integral form. We can provide the student with simulated current sources and have the computer perform the necessary integration. For example, we can model a circular loop of wire. We can tell the student that the loop is perpendicular to a test plane passing through its center and that he can control various properties of the wire—the location of its center, the radius, and the current in the loop (both magnitude and direction). Furthermore, we can allow him to select the point P in the test plane at which the total field $\mathbf{B}(P) = \oint_{\text{LOOP}} \mathbf{dB}$ is to be calculated; that is, he can select the field point.

The particular experimental setting involves the display of the field-point coordinates $(x, y, 0)$, the location of the integration segment, the location of the center of the loop $(x_c, y_c, 0)$, the radius of the loop R, and the current in the loop i. These quantities can be read directly on simple laboratory meters, either with a needle movement or digital readout. The input parameters involved can easily be implemented by potentiometers (or through a digital keyboard in a more complicated hybrid setup). A "rheostat" and reversing switch are particularly appropriate for current control. The values of the components of B can be naturally displayed on "gaussmeters." Finally the test plane of investigation is simulated by an XY plotter, the pen of which is used to locate the desired test point (location of the gaussmeter head); the current element $id\ell$ and the plane of the current loop are simulated by the dot on the face of an oscilloscope (CRT). As the integration proceeds, the dot traces out the loop on the CRT and the B_x, B_y, B_z meters read the magnetic field components due to the path covered up to that point in gauss or webers/meter^2.

The student starts the integration process by putting a three-position mode-control switch into the "run" position; he can stop the process by moving it to the "hold" position which "freezes" the simulation at that point. He can then return to "run" or can start completely over by putting the mode switch in "reset." By alternating between "run" and "hold" he can see quantitatively the contribution to the integral from the small segment covered in between the hold periods. The length of time for the integration can be adjustable although it might be fixed at approximately four seconds. A much shorter time, e.g., one tenth of a second, could be used. One can also allow the student to choose the point on the loop at which he wants the integration process to begin.

The real-time feature mentioned above, although not so critical in this simulation as it is in explicitly time-dependent problems, e.g., dynamics, heat flux, wave motion, etc., is useful to the student in gaining a feeling for the portion of the path of integration that dominantly affects the total field. If he selects the location of his test apparatus (field point) just over the top of the wire, where the wire pierces the test plane, then the portion of the integration path near this point causes the gaussmeters to change appreciably, whereas the contribution from the bottom of the loop is very minor due to the $1/r^2$ nature of the field. This feature can be vividly seen in this simulation and shows up as a time-dependent effect—of the four seconds that it takes the dot on the CRT screen to go around the loop, the "one second of integration" near the top of the loop accounts for almost the total value of B.

To fully appreciate the learning potential pr ·ded by such simulations, one should glance at an introductory text and compare its presentation of the above material with what the student can investigate and learn for himself in an SES environment. (In Halliday and Resnick it is example #8 on page 862.) It was to elucidate, or perhaps replace, such textbook lessons that the CSPEP Project was initiated.

Another example using this same experimental setting, the same from the student's point of view, is the following problem/experiment in which the student can investigate the structure of magnetic fields due to long straight wires and solenoids. Again he must implement his judgments on physical hardware—in this case, the location of the wires, values of currents, number of turns per unit length on the solenoids, etc. After he has constructed the sources that he wants to use, he studies the fields they produce. In this simulation, the components of the total field are presented *continuously* on his gaussmeters as he moves the test instrument around in the test plane. There is no time lag here; the instruments operate as if the meters were actually connected to a Hall-effect transducer and actual fields were being measured (real-time simulation). If the student designs a good experiment, he might discover for himself that the field due to a current in a long straight wire displays a $1/r$ dependence. If he is successful in learning about such field patterns, null points, etc., he will then be able to devise a search procedure to find the coordinates where a wire of unknown location pierces the test plane. This problem was also checked out and used in the pilot course with considerable success.

A wide variety of such simulations for an experimental setting have been considered, and they are by no means limited to the types illustrated by the two examples presented above. Problems like air track or air table simulations, rocket and celestial mechanics, waves and optics, quantum mechanics, and modern and relativistic physics problems are also possible. There are certain features, however, that a phenomenon should exhibit before a simulation approach is justified. A preliminary critique for the selection of phenomena together with a construction paradigm for such simulations appear in the paper "Simulation Techniques in Science Teaching."[3]

PRACTICAL MATTERS

What is necessary to do this kind of simulation? There are three problems to be solved in creating this type of learning environment: (1) obtaining a computer suitable for real-time simulation with high-speed repetitive execution capability, (2) interfacing the computer and the GDCC, and (3) assembling the instruments for the GDCC. The author has made a careful study of the execution times involved in a variety of such simulation programs, particularly those involving differential equations. It appears difficult or impossible to use the GDCC with *only* a time-shared digital computer. In all simulations of any physical significance the digital execution times are a factor of about ten too slow.[4] To achieve the necessary computational capability the machine must be dedicated to the GDCC during the teaching hours, and thus a relatively inexpensive computer should be used. If a digital system is used, foreground/background programming is, of course, permissible and desirable *provided* that student interrupts have priority. Interfacing a digital computer to the GDCC requires analog-to-digital (ADC) and digital-to-analog (DAC) equipment and real-time clock synchronization. Such hybrid items are reasonably sophisticated and expensive, and their use is not recommended for beginners. Finally, one should note that much of the I/O hardware mentioned above is available in any instructional laboratory. It was this type of hardware—meters, oscilloscopes, switches, etc., that was used in the exploratory phase of the work.

The situation and requirements just described lead one to consider using analog computing equipment *as a starting point* for the solution to the above problems. There are several reasons for such a choice. The information (problem variables) in an analog computer is already in the form of continuous voltages suitable for driving standard laboratory measuring equipment. Thus with such a beginning one can completely by-pass the complicated and expensive graphics-interface problem present with an all-digital system. Integrals are performed continously, which increases solution speed. Also such an analog computer and related measuring equipment are very valuable instruments to have available in an intermediate laboratory. They can be put to excellent use for signal conditioning and data acquisition. In fact, one might justify purchasing them on this basis.

Finally, one should note that analog computers are reasonably inexpensive. One can do rather elaborate simulations with a $10,000 to $20,000 machine as in the CSPEP Project. To a certain extent, the cost of an analog computer is related to the complexity of the simulation in a fairly direct way. An analog computer is a "parallel processor." By this is meant that all mathematical operations are performed simultaneously, in contrast to a serial digital machine. This fact also contributes to the high speed of analog computers, but it limits the number of non-linear features that the simulation model may contain since there is a limited number of such computing elements in a machine of this price range.

If one wants to develop this kind of computer learning facility at low cost (1) he must be willing to work hard, be resourceful, and *do it himself;* (2) he should have at least two patch panels for his computer so that a problem under development can be stored while students are working on a completed model (patch panels are the operational equivalent to card decks or magnetic tapes); (3) finally, he must be willing to get his hands on some hardware and laboratory equipment and to do some reading and investigation about it. At present it is probably not possible to "buy into" anyone else's setup or technique. In order to get started one might even borrow a small analog computer, e.g., from the EE Department. Analog computing is easy, and the skill to produce a good simulation program can be acquired with moderate effort.

Table 1 lists the equipment used in the experimental version of the GDCC. The items without approximate prices are not essential at first and were used because it was possible to "scrounge" them. Of the equipment listed, everything needed for the first GDCC was borrowed except the meters. The clarity of display made possible with these meters warranted their purchase. These meters had removable scales which could easily be changed from problem to problem and on which could be written the specific units and scale appropriate for each experiment.

TABLE 1 Equipment List for the First GDCC

2	EAI TR-20 computer @ $12,000	$24,000.
10	API meters with removable scales	250.
10	ten-turn potentiometers	100.
15	switches	25.
1	oscilloscope with chopper unit for y input (Dumont)	250.
1	XY plotter	1,500.
1	digital voltmeter (DVM)	
	parts for frame	75.
	strip chart record (multiple pen)	
	counter (Beckman)	
	signal generators	
		$26,200.

Before giving an explicit example of the design of a simulation model, one should indicate more generally how this type of equipment can be used. There are at least three modes of use for such a facility: (1) individual problem/experiment investigation, (2) lecture demonstration (particularly in conjunction with closed-circuit TV), and (3) real-time film making. Lecture demonstration and film making are particularly needed if large numbers of students are to gain advantage from such a facility. From experience with all three modes, however, the author thinks that (2) and (3) considerably compromise the potential available with such a learning facility.

With respect to the number of students serviced by such a facility, one should note that after its establishment, it can accommodate sixty to seventy students per week in problem/experiment investigations. This schedule allows a pair of students to use the console for two hours on-line each week in a manner not too different from a regular laboratory. The pilot course actually had thirty-five students enrolled in it, all of whom did six individual (pair) studies in six weeks. The experimental facility was operated about seventy hours per week; about two thirds of the time was used by students and the remaining time for hardware and software development.

A note on costs and class size; if one is optimistic, one might make the following cost argument: $30,000 worth of hardware defrayed by 124 student hours/week, i.e., (62 students/week) × (2 student-hours/student), for a 40-week period is $6/student-hour. This figure represents the hardware costs if they are defrayed in *just one year*. One should note the nature of the learning environment created for this amount of money—one not dissimilar to an elaborate laboratory console/experiment station. The modular hybrid learning facility on which the author has been working will have hardware costs approximately two to three times this amount, and it is intended to defray the cost over a five-year period. This leads to "hardware-connect costs" of approximately $2.40–$3.60 per student terminal hour. Fifteen student/teacher contact hours or one semester credit hour at these rates would cost $36 to $54 tuition (for hardware expense). It may be possible to account for the systems development costs through the use of foreground/background programming and the use of the facility after student hours for research.

QUADRUPOLE MASS SPECTROMETER SIMULATION

This simulation is a simplified version of one developed at Marshall Space Flight Center[5]; it will be presented in sufficient detail so that someone not familiar with computing can implement it on a small analog machine. For background refer to (1) D. Martin's article[6] for a brief introduction to analog computing, (2) the programmer's reference manual for the machine used, e.g., the EAI TR-20 Operator's Manual,[7] and (3) one of several books on analog computing. A brief introduction to analog programming in *paperback* form is the book by Zulauf and Burnett.[8] For those not familiar with a quadrupole mass spectrometer the article by Matheison and Harris[9] will prove helpful. Jackson's book[10] contains many examples including Mathieu's equation (QMS) and a model for crossed E & M fields. The physics literature on analog simulation is limited, but the articles on the bouncing ball[11] and on plasma physics[12] should be noted. A valuable source of further material is the Analog/Hybrid Computer Educational Society[13] (ACES), especially the *Transactions* and the *Application Notes*.

Schematic pictures of the QMS instrument are given in Figure 2. Just in front of the rods is an ion source; at the other end of the rods is an ion detector, e.g., an electron multiplier or Faraday cage. The ions travel through the spectrometer or filter lengthwise along the z-axis. As they pass through the region between the rods they experience a force due to the dc and rf electric fields in this region, $-e \nabla \Phi$. These fields are generated by the voltages across the rods which constitute the (approximate) quadrupole surfaces or hyperbolas. The equations of motion for the ions in this region are

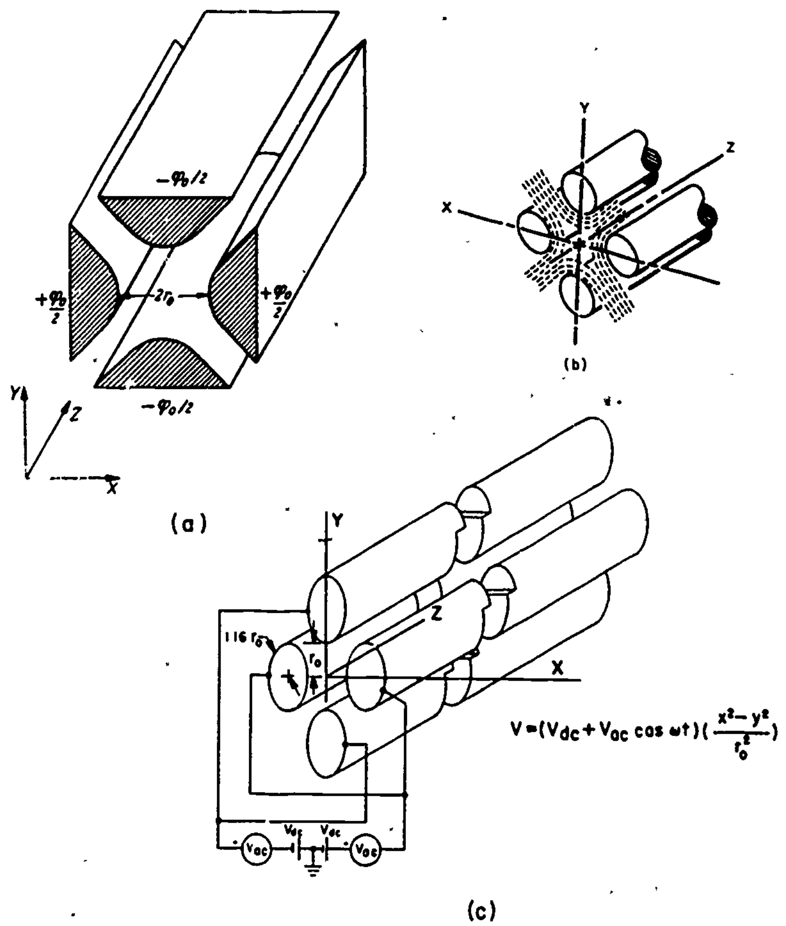


FIGURE 2 Quadrupole Mass Spectrometer (QMS): (a) hyperbolic geometry; (b) approximate quadrupole field lines produced with circular rods, radius $r = 1.16r_0$; (c) potential connections (rf oscillator).

$$\Phi(x, y, z, t) = [(x^2 - y^2)/r_0^2]\ [U + V\cos(\omega t + \varphi)] \tag{1}$$

$$M\, d^2x/dt^2 = -(2ex/r_0^2)\ [U + V\cos(\omega t + \varphi)]$$

$$M\, d^2y/dt^2 = +(2ey/r_0^2)\ [U + V\cos(\omega t + \varphi)]$$

$$M\, d^2z/dt^2 = 0, \quad z(0) = 0.$$

Whether the ions pass through the filter or not is determined by path (trajectory) stability; that is, whether the maximum amplitude of the motion of the ions is bounded in time. Mathematically this is determined by the two coefficients involving the potential parameters U and V; eU/Mr_0^2 and eV/Mr_0^2.

What we would like to do in this simulation is to allow the student to gain some understanding of the equations involved, to relate the variables in the equations to the geometrical properties of the trajectories of the ions, to study the filtering action and the resolution of the instrument, and in general, to become familiar with the "inner workings" of such an instrument. This simulation might also motivate some students to study in detail certain topics in mathematics although only a minimum knowledge of mathematics is needed to gain an understanding of the operation and principles of the instrument from the simulation itself. One needs no previous knowledge, for example, of stability diagrams, for one can easily see the meaning of such diagrams by studying this model. Figure 3 contains photographs of some of the output traces from this simulation and gives a brief idea of what the student can expect to see. These were taken from a real-time film of the operation of the QMS simulation which is presently being prepared in the form of a film loop.

How shall the simulation be constructed? To begin with, it is important for the student literally to see the paths of the ions, something he cannot do in his use of the real hardware instrument. Furthermore, he should be able to manipulate these trajectories with the controls of the (simulated) QMS instrument; for example, to cause ion selection by y-instability. This is quite easy to do. One generates the coordinates of the ion (x,y,z) as functions of time. These functions are obtained by the computer as solutions to the equations of motion and are used as inputs to some laboratory recording device. They can be displayed in pairs (x,y) and (x,z) and/or (y,z) depending on whether two traces can be displayed at once. A slightly more complicated program can make a three-dimensional perspective presentation. Such a display is more artistic than educational, however. More instructionally valuable are phase-plane plots which can be easily displayed also.

A dual-trace memory scope is an ideal display device for use in SES (e.g., Tektronix Type 564). If the computer has a repetitive operation feature, a standard oscilloscope can be used. Care must be exercised in the rf generator part of the simulation, however, if this feature is used. Either a computer independent signal generator is necessary, or a stabilization network on the sine/cosine generator is required. A strip chart recorder can be used for readout since $z(t) = \dot{z}_0 t + z_0$ with $\dot{z}_0 =$ constant; or the traces can be recorded on an XY plotter. Even three needle-movement meters might prove satisfactory for slow trajectory generation. Probably a Heathkit-type oscilloscope would

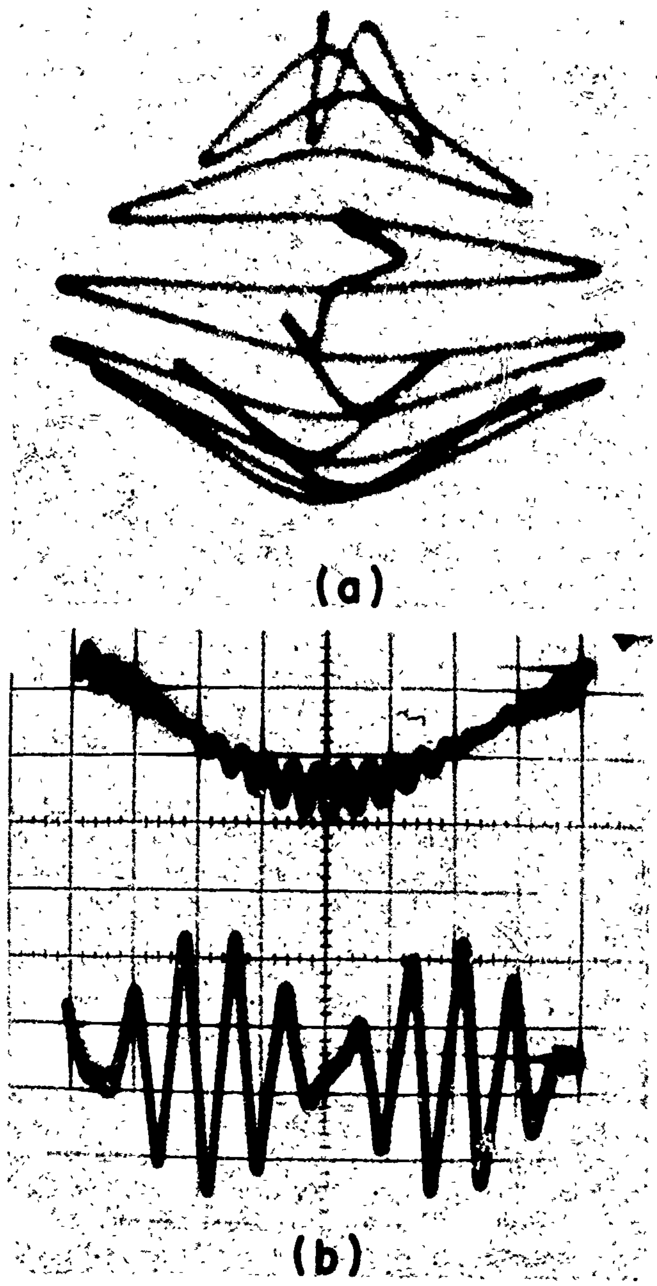


FIGURE 3 Some QMS output curves: (a) ion motion in the xy-plane as seen looking down the quadrupole axis (z-axis); (b) ion motion in the yz-plane (upper curve) and the xz-plane (lower curve).

give the student a better understanding of the ion motion, however, even though the past spot would rapidly fade out.

The physical parameters (instrument controls) of the problem are the initial position and velocity $x_0, y_0, \dot{x}_0, \dot{y}_0$ (position of entrance slit, injection angle and speed), the rf and dc potentials V and U, and the rf frequency and phase ω and φ. It is convenient to read the potential parameters directly on properly labeled meters in kilovolts and megahertz and to control them with potentiometer knobs. The initial positions can be observed directly on the two-dimensional display device, or a meter can be used for these values also. The velocities cannot be observed directly from the trajectories and so must be indicated on meters in cm/microsecond or the equivalent speed for the given mass in electron volts.

Most analog computers have a meter mounted on the front panel that can be switched to various parts of the program to read all the parameters just mentioned. Also the potentiometers for input parameter control can be those on the computer itself. Therefore, for *the first time around*, one only needs some kind of two-dimensional display device, and this item will probably be available with the machine if it is borrowed. I should strongly emphasize, however, that my experience indicates the importance and necessity for some kind of Graphics Display and Control Console as a high-speed student/computer interface. Without the individual labeling of parameters in *appropriate units* on *separate* meters or display devices the student spends considerable time "untangling" the computer hardware. Furthermore, the use of control consoles similar to professional equipment should be an important part of the learning experience for science and non-science students alike.

THE COMPUTER PROGRAM

The first step is to scale the equations (1) of motion to dimensionless form:

$$\ddot{x}/\ddot{x}_{max} = -(b/b_{max})[(U/P) + (V/P)\cos(\omega t + \varphi)](x/x_{max})(b_{max}Px_{max}/\ddot{x}_{max})$$

where $b = 2e/Mr_0^2$ and $P = U_{max} = V_{max}$. A similar equation is written for the y-motion.

$$d^2z/dt^2 = 0 \text{ leads to } z/z_{max} = (\dot{z}_0/\dot{z}_{max})(\dot{z}_{max}/z_{max})\,t + z_0/z_{max}$$

It is these normalized (scaled) equations and variables that are actually implemented on the computer. Dimensionless variables must be used in an analog program since a computer variable (voltage) "analog to" or representing some given problem variable can take on only the values between ± the reference voltage of the computer. This requirement is identical to using fixed-point arithmetic in a digital program and corresponds to fixed-point scaling of a digital simulation model. This range of ± reference is called the dynamic range of the machine and is usually ±10 volts or ±100 volts depending on the machine used. Such analog variables can be thought of as ranging between ±100% of the maximum value of the real-world or problem variable. In the QMS example, the real-world variable x is of interest only if the ion is inside the quadrupole, that is, only if $|x| < r_0$ = field radius. (See Figure 2.) Therefore, if we let $x_{max} = r_0$, a suitable computer variable is $e_1 = (x/r_0)$, since $0 < |e_1| < 1$. This type of scaling called unity scaling, is the one most appropriate for scientific simulation work.

Each mathematical operation indicated in the normalized equations is performed by a specific computer element in the machine and is represented by a box in the flow chart for the problem. The computer manufacturer has developed a variety of such computing elements, each having a characteristic mathematical relationship between its input and output variables; examples are shown in Figure 4. In Figure 4(d), (e) the experiment-time variable is t and n is called the time-scale factor for the problem. In a real QMS the ions travel through the instrument in about 50 oscillations of the rf field or about 10^{-5} second. The computer cannot generate solutions this rapidly even if we wanted it to. If the computer takes, for example, 10 seconds to simulate this flight and if we let τ represent the computer time, then $n = 10^6$ since $\tau = nt$. One automatically has the computer time interval, $d\tau$, proportional to the problem time interval, dt. The term G in Figure 4(d), (e) is called the gain for the element. ($G = 1/T$, T = time constant for the integrator = RC.) G is often equal to one. However, if one increases G, one can make n smaller with a consequent shorter computer time needed to simulate the phenomenon. G can be made large enough in a good computer so that the computer trajectories can be generated in about 10 milliseconds, that is, 100 complete solutions per second. The time scale for this fast case is $n = 10^3$. This procedure is called "changing the time scale of the computer."

When one combines time-scaling and normalized variables, one uses the combination of computer elements shown in Figure 4(e). This combination is important for understanding the implementation of differential equations such as the QMS equations. The other boxes needed in the QMS simulation are also listed in this figure. With the use of this chart one can easily implement the normalized (scaled) equations for this simulation from the flow chart for the problem given in Figure 5.

In connection with Figure 5(a) one should note that C_1 and C_2 are scaling coefficients. C_2' combines the scaling coefficient C_2 and the last bracket in the normalized equations. Also

$$C_1/C_2 = Pb_{max}/n^2G_1G_2 = (bV/\omega^2)[(b/b_{max})(V/P)(n^2G_1G_2/\omega^4)]^{-1}$$

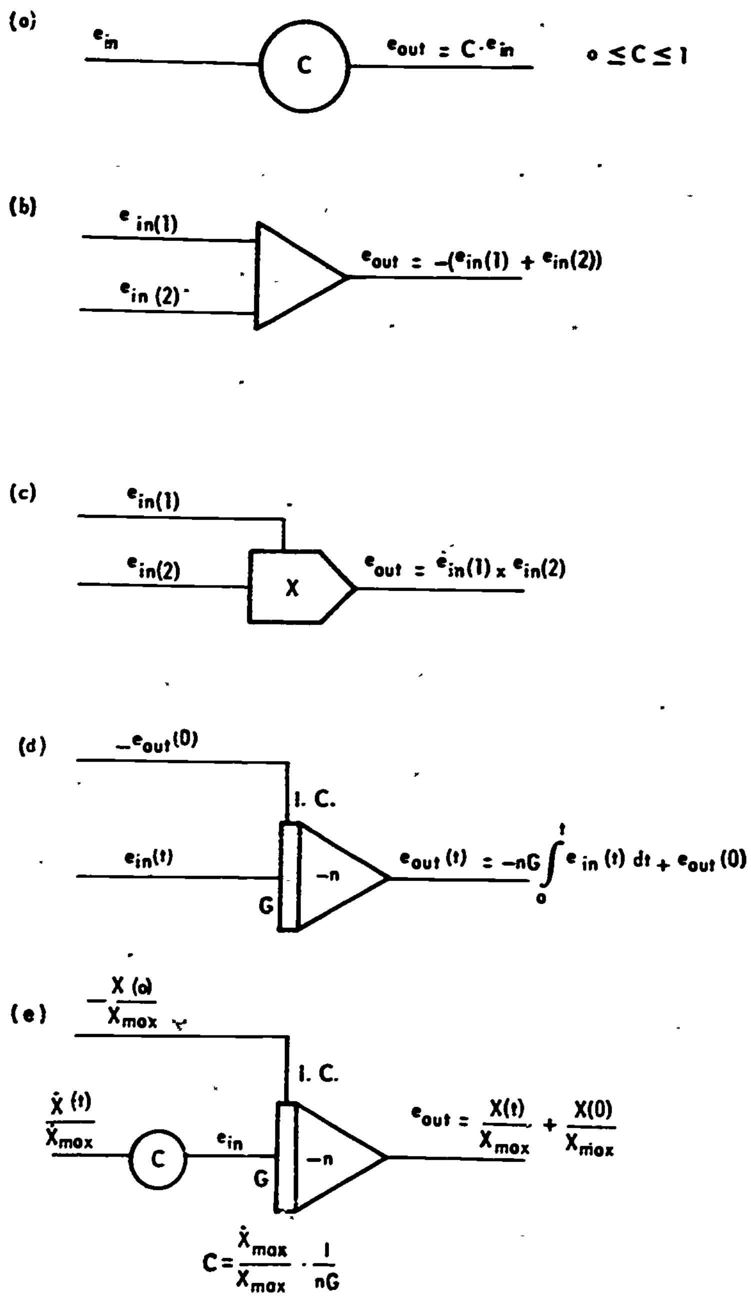


FIGURE 4 Flow-chart symbols used in the QMS simulation and their mathematical equivalents.

If one defines $q = 2bV/\omega^2$ and $a = 2bU/\omega^2$, then ion transmission accurs for $q \cong .706$ and $a \cong .231$ (see section on Mathieu's equation below). Using this value for $2bV/\omega^2$ and assigning $\dot{x}_{max} = \omega r_0$ and $P = 1000$ volts, one gets reasonable settings for the computer coefficients in $nG_1 = \omega$, $G_1 = 0.1$ $G_2 = 1$, $C_1 = 1$ and $C_2 = 0.2000$. If $M_{min} = 40$ amu, then for $M = 90$ amu, $b/b_{max} = (M_{min})/M = 0.4444$ and $V/P = 0.3971$. Note also that a mass of 160 amu, corresponding to a b/b_{max} value of 0.25, is transmitted through the filter at an rf potential value of $V/P = 0.706$, a value numerically equivalent to the q-value for stability in the Mathieu equation. This setting of b/b_{max} is appropriate for mathematical studies. In referring to Figure 5(b), one sees that the parameters ω/nG combine the frequency and the scaling coefficients. If $\omega = n$, $G = 1$, then the circles are eliminated. The loop shown solves the equation $d^2\alpha/d\tau^2 = -\alpha$ with the initial conditions (I.C.) $\alpha(0) = \cos\varphi$ and $\dot{\alpha}(0) = \sin\varphi$. Thus since $nt = \tau$, $d^2\alpha/dt^2 = -\omega^2\alpha$ and

$$\alpha(t) = \cos\varphi \cos\omega t - \sin\varphi \sin\omega t = \cos(\omega t + \varphi).$$

The simulation as implemented in these flow charts has certain simplifications that are undesirable from a pedagogical point of view. For example, some of the experiment parameters such as mass ($b = 2e/Mr_0{}^2$) and phase angle φ must be set on two different coefficient potentiometers—once in the x-equation and once in the y-equation for the mass, and once in $\cos\varphi$ and once in the $\sin\varphi$ for the phase angle. Pedagogically it would be more satisfactory to control one *physical variable* with a *single knob*. In a slightly more sophisticated program this can be done by setting the physical parameters with several multipliers connected to a single coefficient potentiometer. The flow charts in Figure 5, however, contain all the information needed to put this (simplified) program on the computer with the exception of knowing how to implement the boxes for a specific machine. This information is somewhat similar to knowing how to use a keypunch and card reader and is given in the reference manual for the specific machine used.

It is not clear to the author why a newcomer to analog computing finds construction of these boxes on a small analog computer distasteful, whereas he would not hesitate to use the equivalent pieces of technology to implement a digital program. To allay the "fear of wiring" Figure 6 shows just how simply the rf generator is implemented for a TR-10 (an early version of the TR-20). One follows the flow chart and pictorial diagram in the programmer's manual and makes a connection corresponding to a line in the diagram. The next generation of analog computing equipment will eliminate such details, but in the present-day inexpensive machines the boxes must be implemented by hand.

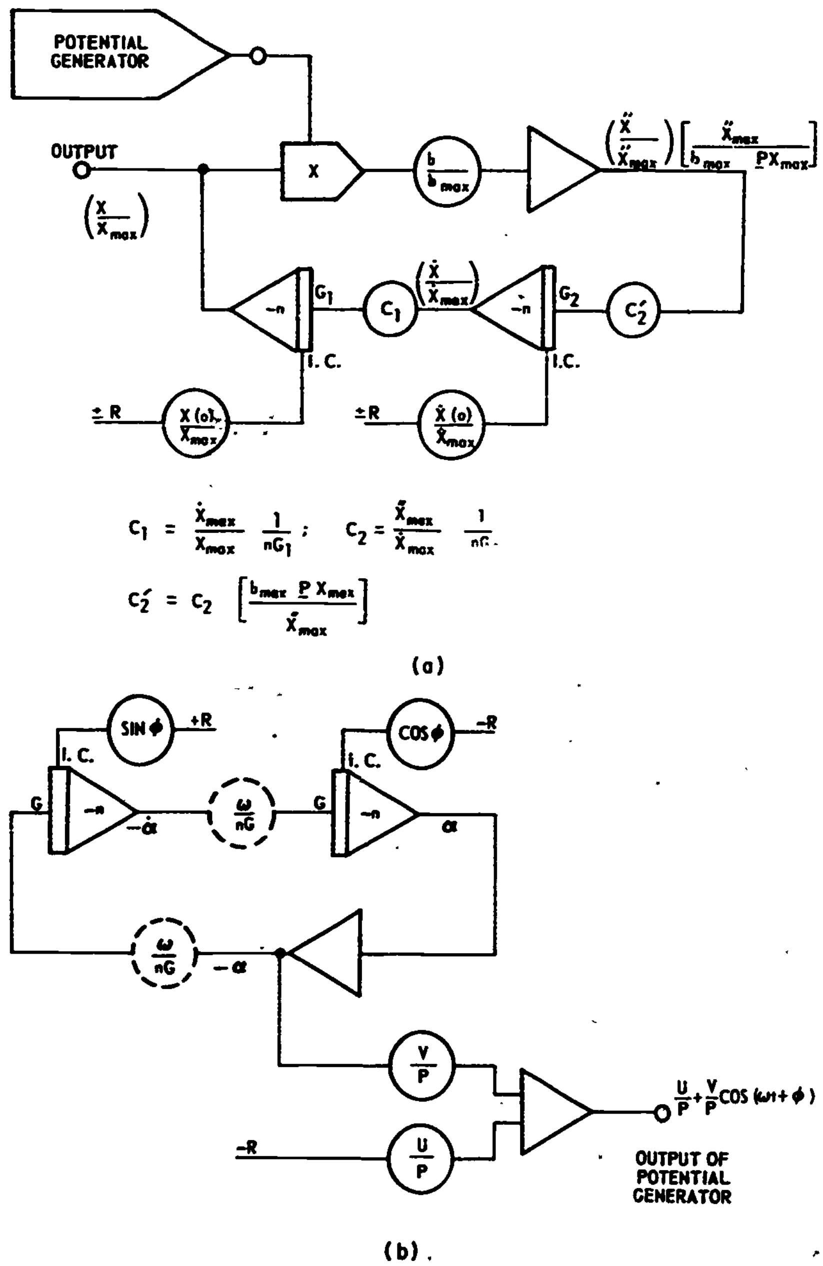


FIGURE 5 Flow charts for (a) QMS dynamics simulator and (b) the QMS potential generator.

It is hoped that the discussion of the QMS problem/experiment has been specific enough so that the reader will be encouraged to start on this simulation as an introduction to the use of such techniques in science teaching. This example was chosen because it is interesting, non-trivial, and of significance in physics. One might note that the x-equation of motion could easily be turned into a driven-oscillator problem. Maybe the reader would rather start on that

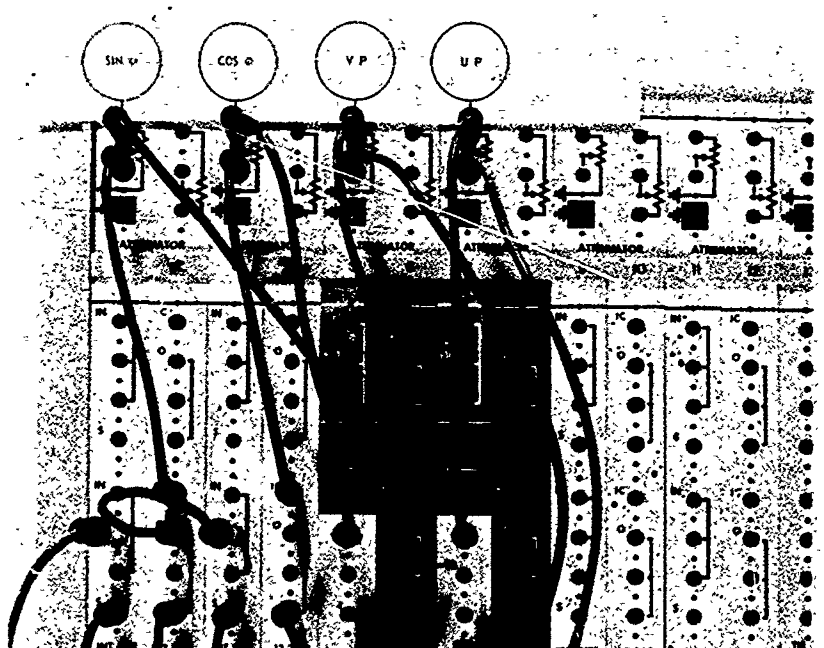


FIGURE 6 "Patchboard" for the potential generator of Figure 5(b).

problem if it is more familiar. It is somewhat different from a physical point of view; the difference in the computer program is trivial.

MATHIEU'S EQUATION

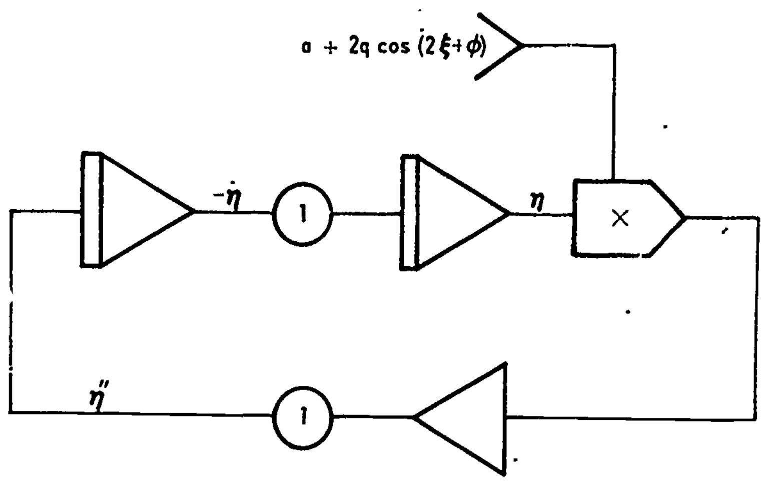
The QMS simulation has been discussed in terms of the *physical* equations of motion. As the reader familiar with the QMS analysis has already recognized, it is actually easier to start with a mathematically equivalent equation, the Mathieu equation. Then, however, the physical meaning of the parameters is hidden. the process of normalization used above is essentially the same as putting the equations of motion into the standard form of Mathieu's equation. Thus, starting with Equation (1) one gets

$$\frac{d}{d(\omega t/2)} \cdot \frac{d}{d(\omega t/2)} \left(\frac{x}{x_{max}}\right) = -(8e/Mr_0^2\omega_2)[U + V\cos(\omega t + \varphi)](x/x_{max}).$$

If one makes the following definitions of ξ, η, a, and q, this equation is converted to the standard form for Mathieu's equation:

$$\omega t/2 = \xi,\ \eta = x/x_{max},\ a = 8eU/Mr_0^2\omega^2,\ q = 4\ eU/Mr_0^2\omega^2,$$

$$d^2\eta/d\xi^2 = -[a + 2q\cos(2\xi + \varphi)]\ \eta.$$



$$n = G_1 = G_2 = 1; \quad \left(\frac{d^2\eta}{d\tau^2}\right)_{max} = \left(\frac{d\eta}{d\tau}\right)_{max} = \eta_{max} = 1$$

FIGURE 7 Flow chart for Mathieu's equation.

Since $\eta_{max} = (d\eta/d\xi)_{max} = (d^2\eta/d\xi^2)_{max} = 1, n = 1 = G_1 = G_2$ and one has the flow chart given in Figure 7, one essentially equivalent to that shown in Figure 5(a). Note the relationship between the scaled time variable τ used in the previous section and the independent variable in the Mathieu equation ξ.

REFERENCES

1. R. M. Brown, *Computer Simulated Problems and Experiments in Physics: Contract Report*, U.S. Army Missile Command. This report discusses the early phase of the project with students, including the pilot course. A limited number of copies are available from the author.
2. D. Halliday and R. Resnick, *Physics: Parts I and II*, John Wiley & Sons, New York, 1966.
3. R. M. Brown, "Simulation Techniques in Science Teaching," in *Proceedings of the Conference on Computers in Undergraduate Curricula*, University of Iowa, Iowa City, 1970.
4. G. A. Korn, "Project DARE: Differential Analyzer Replacement by On-Line Digital Simulation," in *AFIPS Proceedings, Fall Joint Computer Conference, 1969.*
5. R. M. Brown, "Computer Studies of a Quadrupole Mass Spectrometer," in *Research Reports*, Marshall Space Flight Center, Redstone Arsenal, Alabama. 1970.
6. D. C. Martin, "Analog Computers in Science Education," in this volume.
7. *EAI TR-20 Computer: Operator's Reference Handbook*, Electronics Associates, Inc., West Long Branch, New Jersey, 1965.
8. E. C. Zulauf and J. R. Burnett, *Introductory Analog Computation with Graphic Solutions*, McGraw-Hill, New York, 1966.
9. E. Matheison and T. J. Harris, "Quadrupole Mass Spectrometer," *Am. J. Phys. 37*, 1054 (1969).
10. A. S. Jackson, *Analog Computation*, McGraw-Hill, New York, 1960, pp. 142, 149.
11. D. J. Pfaltzgraff, "Analog Simulation of the Bouncing-Ball Problem," *Am. J. Phys. 37*, 1008 (1969).
12. K. Shen *et al.*, "Analog Computer Simulation in Plasma Physics," *Am. J. Phys. 38*, 1133 (1969).
13. *Transactions*, Analog/Hybrid Computer Educational Society, P.O. Box 308, West Long Branch, New Jersey.

Application of Analog Computers in Undergraduate Education

RICHARD M. WILLETT

This paper points out some educational applications of the analog computer which have not been fully exploited. Although analog computers are available on many campuses, the potential use of these computers outside of formal laboratory courses has been largely ignored. This is true in engineering programs, but it is even more prevalent in the sciences. In fact, the analog computer is seldom even seen outside of engineering departments.

We are told that time-sharing terminals are available, which for a modest cost, in effect, bring *digital* computers into the classroom. Then problem solutions can be carried out and demonstrations given to help present material to students. However, many of those functions can be carried out at the present time with desk-top analog computers that can be moved into classrooms and that are already available in laboratories on the campus and hence can be used at *no* additional cost.

For example, introductory physics courses usually include the analysis of falling bodies in a gravitational field, or a projectile following a parabolic path. Solutions of such problems are easily found with an analog computer. Using the repetitive operation mode which repeats the solution many times each second and which is common on modern analog computers, and an oscilloscope for display purposes, a very effective demonstration can be performed to illustrate the physical significance of the various parameters in the problem. Another example might be the solution for the vibrating string. This could include a time-scaled (slow-motion) display to show the effect of initial displacement, tension, elasticity, etc.

In mathematics courses, the solutions of differential equations can be demonstrated showing both transient and steady-state effects by means of repetitive operation. Demonstrations of basic concepts of statistical analysis can be performed using a noise generator and high-speed repetitive operation by simply making repeated runs with randomly varying initial conditions. In electrical engineering studies, the effect of variations in circuit parameter can

Electrical Engineering Department and Computation Center, Iowa State University, Ames, Iowa.

be studied very handily in many problems. The ripple in a power supply with various types of filtering can be readily shown using the non-linear equipment normally present on an analog computer. The effect of changes in the values of circuit parameters on the voltage and current waveforms can then be demonstrated. It is also possible to show the effects of non-linearities in the circuit elements (saturation in an inductor for example), so that the student can appreciate the fact that the linear equations he works with are indeed approximations.

This discussion so far has dealt with using the analog computer as a demonstration device. Another largely neglected aspect is its use as a problem-solving aide in other courses. Many problem assignments must be "watered down" considerably, simply because it is so very tedious and time-consuming to solve them by hand. The digital computer is often used in this application, but the analog computer can handle a large class of problems more cheaply and *more meaningfully* than can the present-day digital computers. The excellent man-machine interface that an analog computer can supply gives the student stronger insight into the significant features of a problem. This is true even when the problems are relatively simple and the solutions are given in the textbook. When complex problems are to be solved, the basic patching can be completed by the instructor and only the parameters of interest need to be varied by the student.

Two examples illustrate the two applications discussed above, i.e., demonstrations and use as a tool in problem solutions. The first example is the application of a small desk-top computer to determine the eigenvalues in the solution of partial differential equations. A typical application of this method is the solution of the waveguide equation. The electric field distribution in a waveguide for TE modes is given by the solution of the equation,

$$\frac{\partial^2 \mathcal{E}}{\partial x^2} = \mu\epsilon \frac{\partial^2 \mathcal{E}}{\partial t^2}$$

This equation can be solved by the method of separation of variables. The solution is of the form

$$\mathcal{E} = [A \sin\sqrt{\mu\epsilon}\,\lambda x + B \cos\sqrt{\mu\epsilon}\,\lambda x]\,(C \sin \lambda t + D \cos \lambda t)$$

In a waveguide, the electrical field parallel to the walls of the waveguide must go to zero, so $\mathcal{E} = 0$ at $x = 0$ and at $x = L$ where L is the width of the waveguide. From these boundary conditions, $B = 0$ and $\sqrt{\mu\epsilon}\,\lambda L = n\pi$, where n is an integer. When $n = 1$, the corresponding λ is the cutoff frequency of the waveguide. As long as ϵ is constant the solutions are easily obtained and are given in any standard text such as Ramo and Whinnery. However, if ϵ is a function of x, the solution is extremely difficult to find and the cutoff frequency or the frequencies of higher modes cannot be readily determined. An air-filled waveguide with a dielectric slab inserted is an example of such a problem.

The analog computer is patched in such a way that the function $\sin \sqrt{\mu\epsilon}\,\lambda t$ is generated. The parameter ϵ must change value when $x = x_1$, where $0 < x_1 < L$ and at x_2, where $x_1 < x_2 < L$. This simulates a dielectric slab in the waveguide, shown in Figure 1. Consider x to be proportional to t, then $\sin \sqrt{\mu\epsilon}\,\lambda t = \sin \sqrt{\mu\epsilon}\,\lambda x$, and t_1, t_2 and t_L are proportional to x_1, x_2 and L, respectively. At $t = t_1$ (i.e., $x = x_1$), the value of ϵ changes from ϵ_0, the value in air, to ϵ_1, the value in the dielectric material. Then when $t = t_2$ $(x = x_2)$, ϵ returns to its original value. This switching is done by using electronic comparators and electronic switching in the computer. The value of λ is then adjusted until $\sin \sqrt{\mu\epsilon}\,\lambda t_L$ is equal to zero, since the function must be zero when $x = L$. The frequency of cutoff is then found from the value of λ. This assumes of course that $\sin \sqrt{\mu\epsilon}\,\lambda t$ does not go through zero between the boundaries. If it does, then λ will correspond to some higher frequency mode rather than the cutoff frequency. The wave shape of $\sin \sqrt{\mu\epsilon}\,\lambda t$ is also shown in Figure 1 with the λ adjusted to the cutoff frequency. This method could be used when several different dielectric materials are present and can be applied to any partial differential equation that can be solved by separation of variables.

The second example uses a somewhat larger computer which includes a resolver and some logic and more switching capability. Introductory courses in automatic control require some use of the root-locus method of factoring the characteristic equation. This is very time consuming, and as a result the instructor often does not assign problems of any consequence. The students

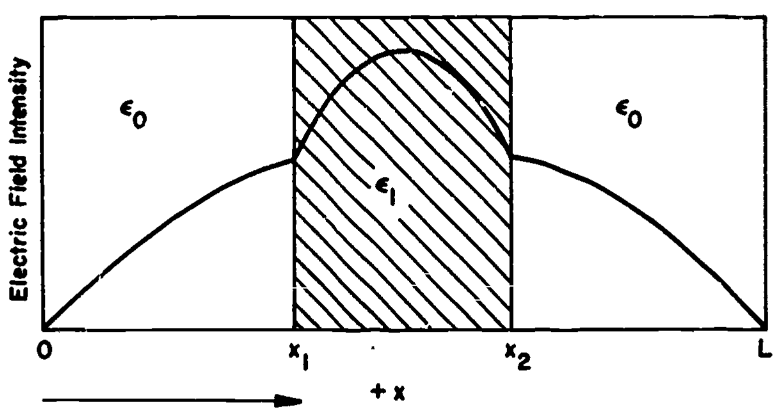


FIGURE 1 ℰ-field with dielectric slab in a waveguide.

in these courses may never work with problems from the "real world" since these are usually too complex and it takes too much time to plot the root-locus. This analog computer program takes much of the agony out of these problems and therefore allows more meaningful problems to be attacked.

The root-locus method is based on the equality $1+G(s) = 0$ or $G(s) = -1$, where $G(s)$ is the "open loop" transfer function. $G(s)$ is of the form $\Pi_k(s + \alpha_k)/\Pi_\ell(s + \beta_\ell)$, where α_k, β_ℓ and s may be real or complex. The root-locus is plotted using the criterion that the complex number $G(s_0)$ (where s_0 is some value of s) must have an angle of 180° if s_0 is to be a point on the locus. The real and imaginary parts of the α_k and β_ℓ are set on potentiometers as are the real and imaginary parts of s_0. The differences $\alpha_k - s_0 = \Delta\alpha_k$ and $\beta_\ell - s_0 = \Delta\beta_\ell$ are found for each pole and zero, and the angles of the vectors $\text{Re}\{\Delta\alpha\} + j\text{Im}\{\Delta\alpha\}$ and $\text{Re}\{\Delta\beta\} + j\text{Im}\{\Delta\beta\}$ are found for each pole and zero (see Figure 2). The sum of the angles of the zeros and the sum of the angles of the poles are found and the difference taken. When this difference of the sums equals 180°, the value of s_0 that was used in the computation is a point on the locus.

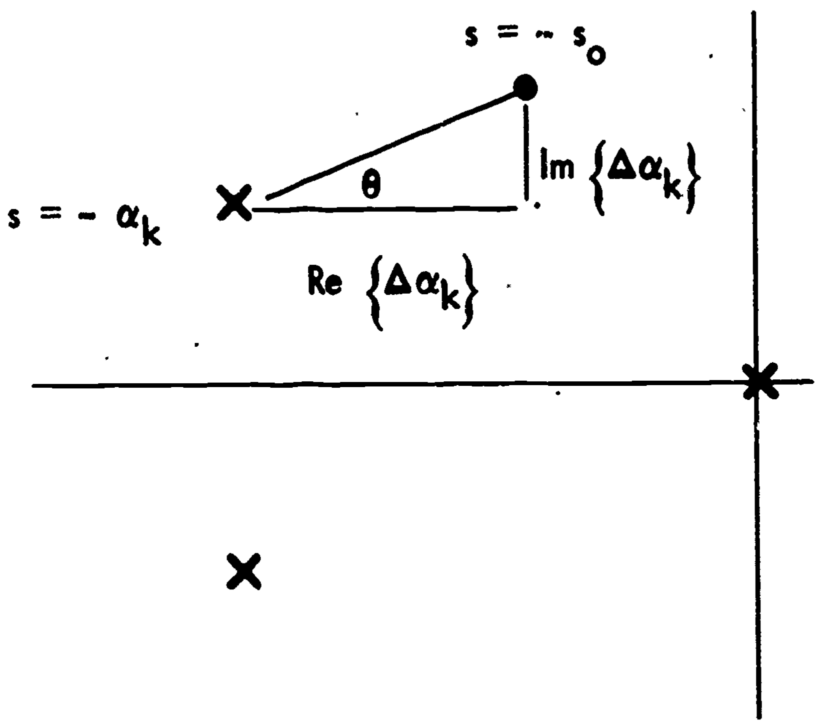


FIGURE 2 *S*-plane diagram showing θ and $\Delta\alpha$.

The calculation of the angle θ for each pole and zero is done in a resolver. The program requires only one resolver since the output for each α_k and β_ℓ is found sequentially by means of a switching sequence, and the angle found for each pole or zero is used to increment or decrement an analog accumulator. The output of this program is displayed on an XY plotter. The effect of changes in various parameters can be determined very quickly and appropriate compensation schemes analyzed. The program presently does not compute the loop gain, but this feature could readily be added. Detailed patching diagrams of these two programs are shown in Figures 3 and 5.

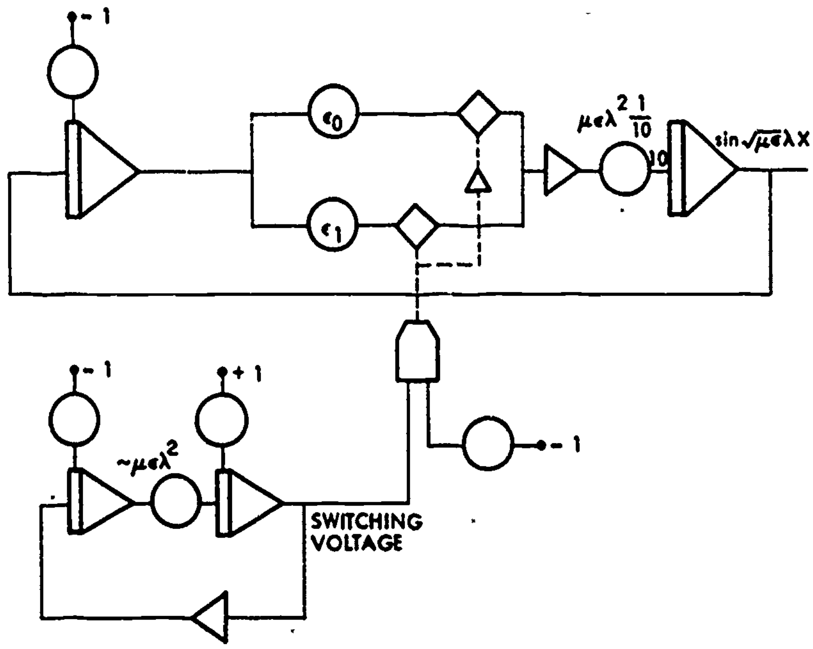


FIGURE 3. Patching diagram for E-field simulation.

Figure 3 is the patching diagram for the demonstration showing the effect of placing a dielectric slab in a waveguide. The diamond-shaped symbols represent electronic switches driven by logic signals.* When the dielectric constant ϵ equals ϵ_0 (see Figure 1), the upper switch is closed and the three amplifiers operate a sine-wave oscillator. When $\epsilon = \epsilon_1$, the upper switch opens and the lower switch closes so that the loop gain is changed. This results in a change in the frequency of the oscillation. When $\epsilon = \epsilon_0$ again, the lower switch opens and the upper one closes to return to the original frequency. All this switching takes place during one half-cycle of the output of the

*Also see D. Martin, "Analog Computers in Science Education," in this volume.

oscillator. The potentiometer preceding the last integrator is manually adjusted to make ℰ, the electric field, go to zero at the wall of the waveguide.

The logic signal used to perform the switching at the proper times is generated by the lower three amplifiers and the analog comparator. The three amplifiers generate a sine wave of about the same frequency as that produced to simulate the ℰ field. The output is compared with some constant negative voltage. When the sum of these two voltages exceeds zero, the output of the comparator becomes a logic "one" and the switching described above takes place. When the sum of these voltages drops below zero, the output of the comparator goes to a logic "zero" and the switches return to their original state. By changing the amplitude and phase of this second sine wave and the dc voltage, the width and position of the dielectric slab can be changed. Higher frequency modes can also be found and their configurations shown. Figure 4 shows the cutoff frequency (top curve) and three higher frequency modes, as they were obtained from the simulation. It is clear from the figure that the dielectric slab is not exactly centered in the waveguide, since the form is not symmetrical with respect to the vertical axis.

This simulation was performed on a TR-20 desk-top analog computer using an oscilloscope to display the results. Both the computer and the oscilloscope can be put on a cart and wheeled into a classroom for use during a lecture. The total time required to patch this problem and prepare for a demonstration is less than one hour. If removable patch panels are available, the

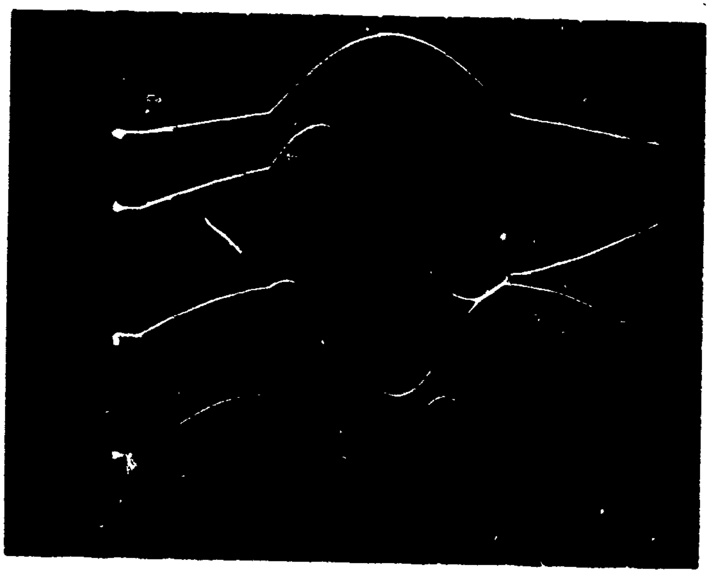


FIGURE 4 ℰ-field simulation results for four different frequencies.

patching could be saved and used for several demonstrations which may be days or weeks apart.

Figure 5 shows the patching diagram of the program used to make root-locus plots. The upper six potentiometers on the left-hand side of the figure are used to set in the values of the real parts of the poles and zeros. In this particular case, there were five poles and one zero. The lower six potentiometers are used to set in the imaginary parts of the poles and zeros. The potentiometers marked A and B are used to set in real and imaginary parts of the trial locations for the roots of the s-plane. When the logic signal S_1 goes to "one," Δx_1 and Δy_1, the real and imaginary parts of the vector from the first pole to the trial point, are found. These are then resolved into a magnitude and an angle θ_1. The magnitude is not used, but the angle is stored in the analog accumulator composed of the first two-track/store amplifiers. When S_2 comes high, Δx_2, Δy_2 and the next angle θ is calculated and the accumulator increments by an amount proportional to θ_2. The process is repeated for all the poles. When S_6 comes high, the angle is computed as before, but since the sixth input is a zero, the angle must be subtracted rather than added to the value contained in the accumulator. To accomplish this, S_6 also drives a pair of electronic switches that reverse the sign of the input to the accumulator. S_6 also switches another track/store amplifier in such a way as to hold the final value of the summation of all these angles so that the final value can be more easily observed.

The logic pulses, $S_1, S_2, \ldots, S_6$ are formed by shifting a single bit through one eight-bit shift register (actually two four-bit registers in cascade). These logic pulses are also modified and combined in an "or" gate and then used as the input to a monostable multivibrator. The result is a narrow pulse output from the multivibrator each time one of the logic pulses appears. These narrow pulses are then used to control the analog accumulator and switch the computer mode. Switching the computer to the Initial Condition (IC) Mode at the start of each pulse prevents overloading of the resolver.

The integrator, potentiometer, comparator, digital differentiator and monostable multivibrator in the lower right of the drawing comprise a voltage-controlled oscillator. The output is a pulse train used as a clock to control the logic. The clock rate can be varied by changing the pot setting.

The trial root position (x_0, y_0) is displayed directly on an XY plotter. The "open-loop" pole-zero locations can be plotted on the same paper being used to record the results. The trial values of x_0 and y_0 are adjusted until the sum of the angles is 180°. This corresponds to 9.0 V out of the last track/store amplifier. Output of 27 V, 45 V or any odd multiple of 9 V will also be acceptable. When the correct angle has been found, the pen on the XY plotter is dropped, which leaves a dot on the paper. The values of x_0 and y_0 are then changed and a new point found. With a little experience it is

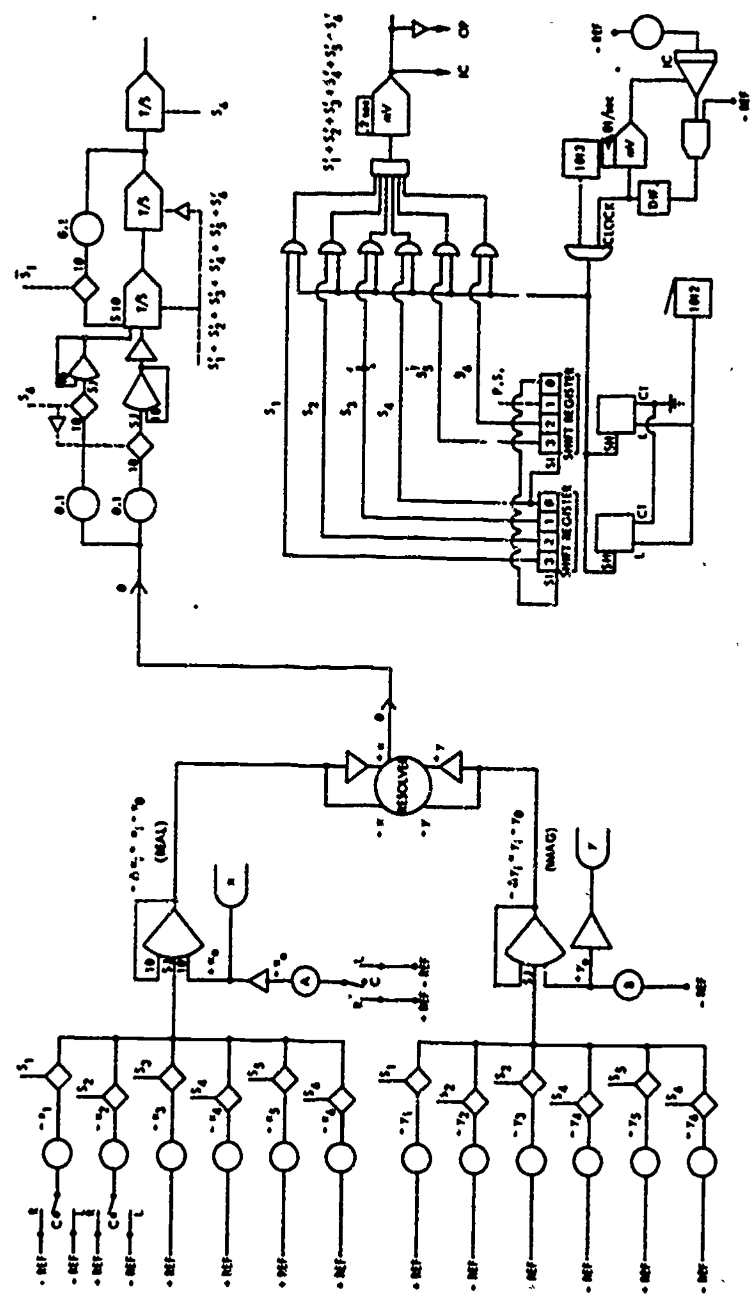


FIGURE 5 Patching diagram results for root-locus plotter.

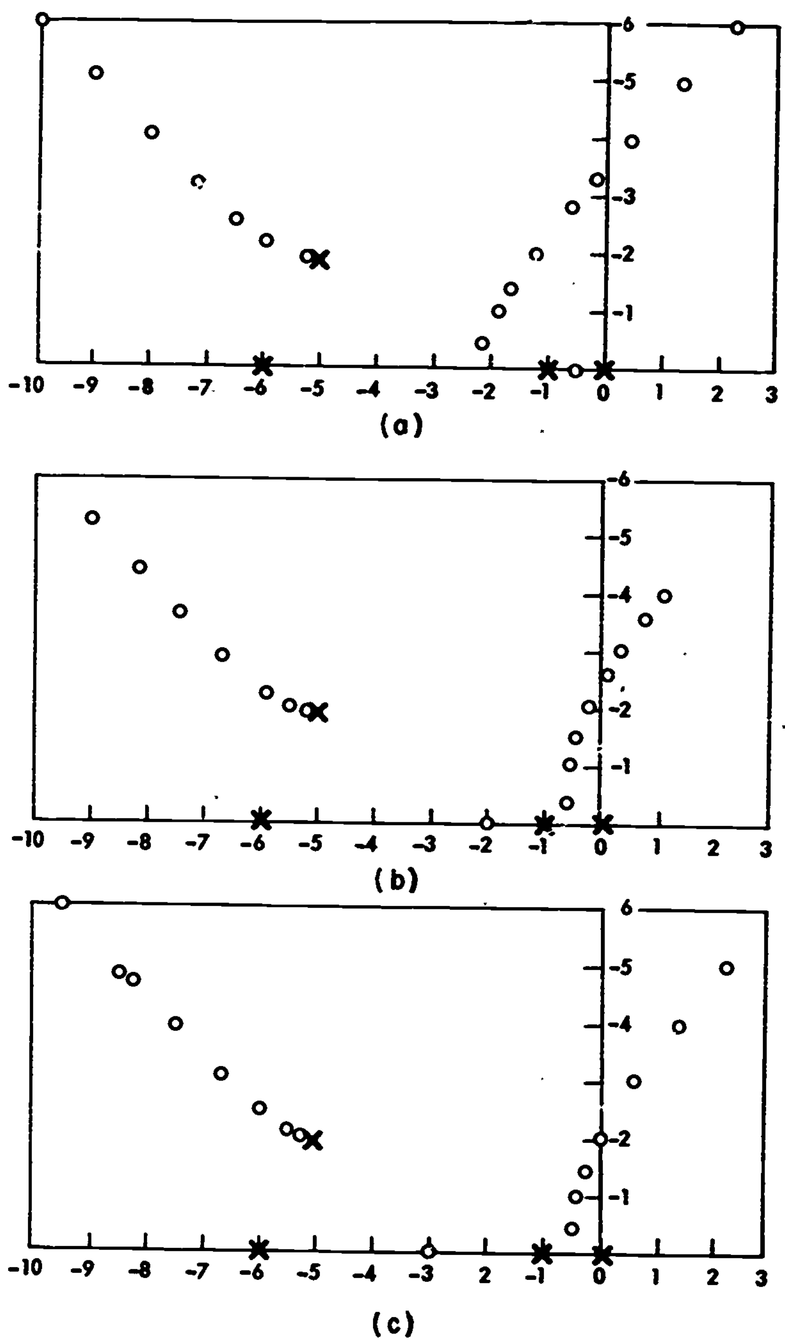


FIGURE 6 Root-locus plots for $G(s) = s + \alpha/[s(s+1)(s+5 \pm 2j)(s+6)]$; (a) $\alpha = 0.5$, (b) $\alpha = 2.0$, and (c) $\alpha = 3.0$.

possible to plot a locus in a few minutes. The most time-consuming task is changing the paper on the plotter. Figures 6(a), (b) and (c) show the results of this program for a particular transfer function

$$G(s) = s + \alpha/[s(s + 1)(s + 6)(s + 5 + j2)(s + 5 - j2)]$$

where α has the value 0.5, 2.0, and 3.0 in Figures 6(a), (b) and (c) respectively. The student using this program does not need to understand the details of the simulation. He enters only the pole-zero locations and the trial root positions and observes the resulting angle. He then varies the trial location and drops the pen on the plotter when the proper locations are found.

The initial set-up time for this problem is fairly long; however, using removable patch panels allows the program to be stored between uses. To be most effective, the computer could be set up for this problem one afternoon each week, for example. Instructors could then schedule their assignments accordingly.

System Analysis and Analog Computer Simulation of Pollution Problems

THOMAS T. LIAO

INTRODUCTION

The magic word for the seventies is "environment." It is important that people become aware of man's pollution of his environment, but it is more important to find ways of coping with the problems. Systems analysis, a logical approach to decision-making, can be used to help man manage the quality of his environment.

Environmental problems in general have the following characteristics: (a) they are multivariate; (b) interactions between components are more important than the components themselves; (c) time variation is a dominating factor; (d) decisions must be made under a high degree of uncertainty, placing great dependence on statistical inference and probability; (e) economic efficiency is an important consideration; (f) solution depends heavily upon quantification of the elements of a system and on optimization of the system response. This study is designed to illustrate how the systems approach to decision-making can be used for water pollution policy problems.

Decision-making via systems analysis involves four basic elements:

Model: the model is a descriptive or functional representation of the problem we are concerned with.

Criteria: the criteria as the goals or objectives of the decision-making problem.

Constraints: constraints are added factors that must be taken into account in the solution of the decision problem or factors that limit the range of permissible solutions.

Optimization: once the problem is formulated (the model), we decide what we really want (the criteria) and statements exist as to what is permissible (the constraints), we are ready to attempt to find the best or optimum solution. Figure 1 shows the dynamics of the decision-making process and how the interaction of these

Engineering Concepts Curriculum Project, Polytechnic Institute of Brooklyn, Brooklyn, New York.

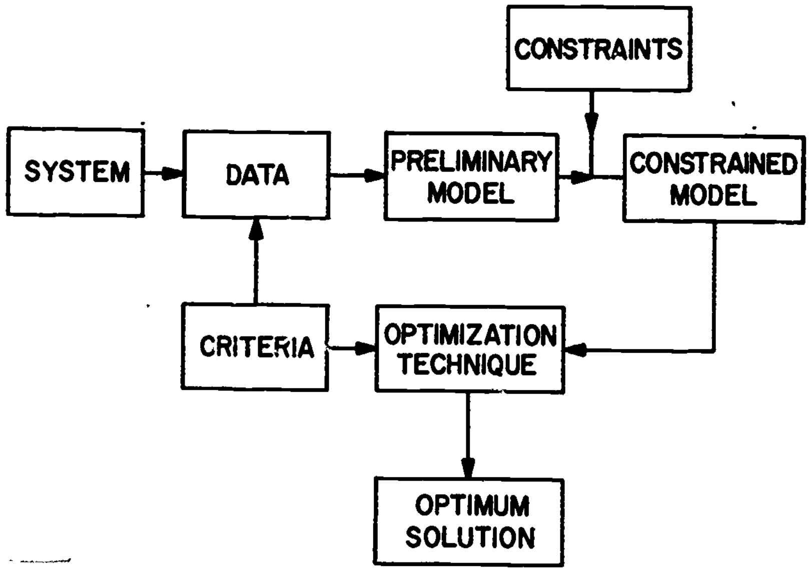


FIGURE 1 A systems approach to decision-making.

four elements can result in an optimum solution. This decision-making process is used by students in a new course "The Man Made World," developed by the Engineering Concepts Curriculum Project (ECCP).

Before data can be collected to develop a preliminary model of a system, the major criteria and the system itself must be identified. Standards for clean water can specify dissolved oxygen, floating solids, color and turbidity, coliform bacteria, taste and odor, temperature, acidity (pH level), radioactivity, and others. Since the type of pollutant varies from one industry and municipality to another, including at times shipping and agricultural wastes, one should pick an indicator that will give the general picture of existing water quality. The best indicator is the amount of dissolved oxygen in the water. In addition to supporting aquatic life, oxygen is directly related to the amount and character of unstable organic matter in the water. Dissolved oxygen is also a good indicator because, in addition to monitoring the level of sewage, it also indirectly monitors nutrient (phosphates) and thermal pollution.

Since we have chosen dissolved oxygen (DO) as our indicator of water quality, it is essential that we develop models for understanding and predicting how the level of DO changes with changing conditions in the management area. Oxygen content of water is regulated by many factors. It is used up by oxidation of sewage, dead plants and inorganic compounds as well as by plant respiration. In a complex model we would want to include all the factors that affect oxygen content. However, we will be less ambitious and only consider two major factors, replenishment by re-aeration and usage by biochemical oxygen demanding (BOD) substances in our model. We will first analyze mathematical models for the oxygen content system, and then we will use them to develop an analog computer simulation (see Appendix A for a description of the analog computer used by ECCP students) of the system.

SIMULATION MODELS OF OXYGEN CONTENT

In order to formulate an oxygen content model for simulation by the analog computer we must understand the rate of decomposition of wastes, the rate of replacement of oxygen in the water, and how the amount of oxygen is affected by these rates. We shall study these questions by doing the following three experiments:

Experiment 1 – Model of the Decomposition of Waste;
Experiment 2 – Model of the Replacement of Oxygen;
Experiment 3 – Improved Oxygen Content Model.

In these experiments we will be using the conventional unit (milligrams/liter) for our measurements of waste and oxygen concentrations.

Experiment 1 – Model of the Decomposition of Waste

In order to develop a waste decomposition model, we must identify the factors that affect the rate of change of waste in the water, dW/dt. One factor is the amount of waste that is in the water at a certain time (W). Other factors are the temperature of the water and the chemical composition of the waste, which can all be represented by a constant C_w. In general, it has a value between 0.25 and 0.75. Scientists using measurements from natural waterways and laboratory water tanks have found that:

$$dW/dt = -C_w W$$

In this experiment we are interested in knowing how long it will take for 50 mg of waste product (W_0) to decompose in a liter of water. Let one second of the analog computing time represent one day of real time and let one volt represent 10 mg/liter. Using Figure 2,* simulate the decomposition of waste on the analog computer. The questions we ask are

1. How does changing the value of C_w affect the decomposition process?

*Editor's Note [RB]: see D. Martin, "Analog Computers in Science Education," in this volume, for discussion of analog flow-chart conventions.

2. For $C_w = 0.5$, how long will it take to decompose 50 mg/liter of waste?

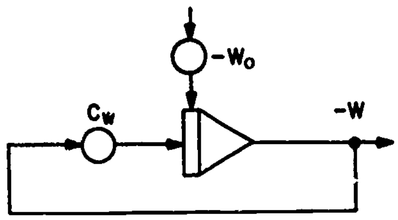


FIGURE 2 Model for decomposition of waste.

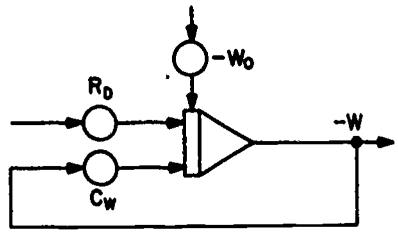


FIGURE 3 Improved decomposition model.

In the previous model, we assumed a certain amount of waste being decomposed without new waste being added. In reality this is usually not the case. Let us improve our model by assuming that there is a constant rate of waste being dumped into the water (R_D). Our mathematical model would become (see Figure 3):

$$dW/dt = R_D - C_w W$$

To perform this part of the experiment:

3. Start by assuming that R_D is 10 mg/liter/day (1 volt). Use the simulation of this improved model and study the effect of varying C_w.
4. Now keep C_w at 0.5 and study the effects of varying R_D.
5. Will the wastes ever decompose completely?
6. What controls the amount of waste that remains in the water?

Experiment 2 — Model of the Replacement of Oxygen (Aeration Process)

Before studying how the decomposition of waste affects oxygen content (Experiment 3), we must first study how oxygen is replaced in the water. From studies it has been found that the rate at which oxygen is replaced dA/dt, is proportional to the difference between the amount of oxygen contained in the water at any time (A) and the maximum amount (A_{max}) that the water can hold. Mathematically the rate of replenishment can be expressed as:

$$dA/dt = C_A (A_{max} - A)$$

where C_A is a coefficient that depends on the turbulence of the water. A_{max} depends on the temperature of the water. Tables 1 and 2 give typical values of these two constants.

TABLE 1 Typical Values of C_A

Large ponds	0.4
Large lakes	1.0
Slow moving streams	1.5
Rapidly moving streams	3.0

TABLE 2 Typical Values of A_{max}

Temperature (°F)	32	41	50	59	69	77
A_{max} (mg/liter)	15	13	11	10	9	8

Use Figure 4 (which is based on $R_A = C_A (A_{max} - A)$, to simulate the replacement of oxygen for different values of C_A and A_{max}:

1. Assuming that we have 10 mg/liter (1 volt) of oxygen at the start (initial condition of integrator at 1 volt), and using A_{max} = 15 mg/liter, study the effects of varying C_A (values from Table 1).
2. Next, keep C_A = 1, and study the effects of varying A_{max} (values from Table 2).

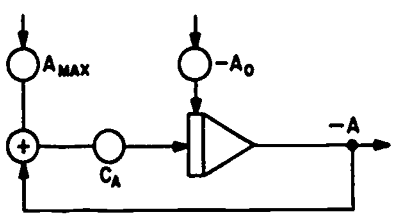


FIGURE 4 Model for replacement of oxygen.

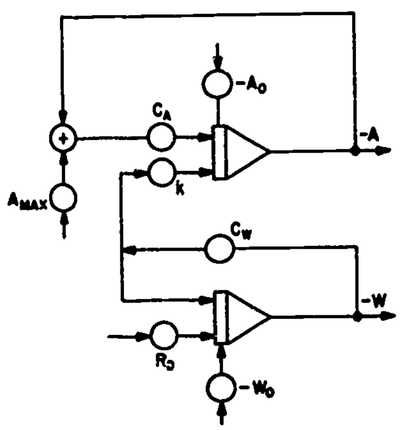


FIGURE 5 Model for Monitoring oxygen content.

Experiment 3 – Improved Oxygen Content Model

Although oxygen is constantly being replenished in the water by air, it is also being used up in the decomposition process. We would now like to combine the models studied in the previous two experiments and generate a model that will monitor the oxygen content at any time. Remember that the rate of replacing oxygen dA/dt was mathematically expressed as:

$$dA/dt = C_A (A_{max} - A)$$

If we want to include the use of oxygen in the decomposition of waste, the mathematical model will become:

$$dA/dt = C_A (A_{max} - A) - A_w$$

where A_w represents the rate of use of oxygen by the wastes. This rate of usage directly depends on the rate of decomposition. Therefore A_w is directly proportional to $C_w W$. Mathematically:

$$A_w = k\, C_w W$$

where k is a coefficient that represents the oxygen requirement (oxygen demand) for a particular waste product. The final mathematical model which we will use for our simulation of the oxygen content model is then:

$$dA/dt = C_A (A_{max} - A) - k\, C_w W$$

We use Figure 5 to simulate oxygen content model on the analog computer. We now want to use the model to answer certain questions that are related to the oxygen content in the water.

Based on the fact that fish begin to die when the oxygen content approaches 5 mg/liter (0.5 volt), the student must answer the following questions (assuming that $k = 1$):

1. At what rate can waste be dumped (R_D), if you want to keep the fish alive? Assume that $W_0 = 50$ mg/liter (5 volts), $C_w = 0.5$, $A_{max} = 15$ mg/liter (1.5 volt, value then $T = 32°$F), $C_A = 1.0$ (value when body of water is large lake) and $A_0 = 10$ mg/liter (1 volt).
2. What is the maximum allowable value for W_0, if we want to keep the oxygen content above 5 mg/liter, if R_D is zero (no dumping of waste after initial amount), $C_w = 0.5$, $C_0 = 1.0$, $A_{max} = 15$ mg/liter and $A_0 = 10$ mg/liter?

3. Study the effect of changing C_w, when W_0, A_{max}, A_0, and C_A are predetermined (choose values for A_{max} and C_A from Tables 1 and 2 from previous experiment). You might want to get data for W_0 and A_0 for a body of water near where you live, or else just choose some representative values. For some situations you may find that the oxygen content goes below zero (negative). This shows that our model has limitations and will only work within a range of values.
4. What happens to the oxygen level when k (oxygen demand) is varied? How is this related to the amount of waste treatment?

The preceding experiments allow students to develop an understanding of the dynamics of dissolved oxygen content. The significant variables can be changed, and the corresponding oxygen levels can be monitored. Graphs of W vs. t and DO vs. t can be plotted simultaneously so that various pollution policies can be evaluated in terms of oxygen content. Once an accurate model is obtained for a particular water management region, techniques such as linear programming can be used to incorporate technological, political and other constraints so that an optimum solution can be obtained. The criterion function to be optimized might be a certain level oxygen content at the cheapest cost.

CONCLUSIONS

One cannot argue with the concept held by science teachers, in general, that their students should be exposed to real-life experiments whenever this is possible. Many times, however, the student cannot perform a specific experiment and, as a consequence, is robbed of worthwhile learning experiences. The systems approach to problem solving and computer simulations (analog as well as digital) can be significant tools for science education. Complex problems which up to now have been treated superficially can be studied quantitatively. The availability of computer simulations should increase substantially the number of experiments that the student may conduct.

Simulations are appropriate substitutes for real-life experiments when equipment is unavailable due to expense or complexity, or measurement and analysis is, for various reasons, impossible or too time-consuming. Furthermore, simulations can be used to model systems that do not as yet exist or, perhaps, may never exist, but are of great mathematical interest. The objectives of simulation should be (1) to improve the student's understanding of subjects inadequately treated in conventional laboratories, (2) to provide opportunities for learning by observation rather than vicariously, and (3) to permit presentation of topics previously too difficult, mathematically, for treatment in the course.

APPENDIX A

The analog computer is a device that can perform certain mathematical operations on signals that are fed into it. Large sophisticated computers can add, subtract, scale (multiply a signal by a constant coefficient), integrate, multiply two signals, square, take the square root, divide and perform certain trigonometric operations. The AMF (American Machine and Foundry Corp.) unit used by ECCP students can only perform the operations of addition, subtraction, scaling and integration. However, this is adequate for us to solve a wide variety of engineering problems. AMF makes two models of analog computers for use in ECCP classes (demonstration and student models). Both consist of three summing units (with scalers), two integrators, a built-in power supply, and a center zero dc meter. The demonstration unit comes with a transparent projection meter and costs $430, while the student model costs $315.

REFERENCES

1. R. Dorfman and H. D. Jacoby, *A Model of Public Decisions Illustrated by a Water Policy Problem*, Discussion Paper Series, Harvard Institute of Economic Research, Cambridge, Mass., 1969.
2. *The Man Made World*, Engineering Concepts Curriculum Project (ECCP), McGraw-Hill, New York, 1968.
3. *Engineering*, Cornell Quarterly, College of Engineering, Cornell University, Spring 1967 and Winter 1968.
4. G. M. Fair and J. C. Geyer, *Water Supply and Waste Water Disposal*, John Wiley & Sons, New York, 1954.
5. *Water and Air Resource Management*, IBM Scientific Computing Symposium, 1968.
6. L. C. Urquhart, *Civil Engineering Handbook*, McGraw-Hill, New York, 1959.
7. R. L. Wilson, *Simulation of Oxygen Dynamics Relative to the Purification of Fresh Water*, Electronic Associates Inc., Bulletin ALAC, 65056.

V

COMPUTER-SUPERVISED INSTRUCTION

Introduction

During the 1967-68 academic year a completely computer-administered physics course for non-scientists was given at Florida State University. Some 40 students were chosen at random from a class of 750. Their interactive terminal was the CRT of the IBM 1500 system; input and output were entirely alphameric; there were no lectures and no live demonstrations. Students were instructed by the computer when to listen to tapes and to watch film loops, and proceeded at their own pace. The organization, economics and outcome of this experiment is described in the paper by Kromhout *et al.*

While the Florida experiment was essentially an isolated one, though very complete, the PLATO project of the University of Illinois has been in continuous operation for the past decade, using and generating a remarkable variety of materials. These are illustrated in the article by Bitzer *et al.*, who make the important point that their materials are authored by discipline-oriented academicians not by programmers. So versatile has PLATO proved in its manifold uses, that we have availed ourselves of editorial license in order to include some examples of materials from peripheral and non-scientific education such as maternity nursing, demography, and elementary school education. Also discussed is the TUTOR language used with the PLATO system.

One common approach to tutorial computer-assisted learning is through the development of *author languages* whereby the teacher may readily construct teaching programs. Such a language must allow him to enter text and questions without worrying about the format and specify both right and wrong answers and the prescribed responses given to each, and what to do in the case of unanticipated wrong responses. It should have the ability to count errors, to store responses, to grade, and to provide instructors with a permanent record of performance. Also desirable is the provision of computational facilities through the interactive terminal, so the student can, if desired, actually solve problems by running short programs in the course of answering questions.

Over thirty such languages presently exist, but they are generally specific to certain machines and systems, expensive to develop and difficult to export. Another approach is to work within one of the powerful existing programming languages such as FORTRAN, PL/1, APL, SNOBOL, and SUPER-BASIC. One such attempt is the development of a language, DITRAN, for the construction of tutorial programs by means of "templates." This method is described in the paper by Sherman. It is a method in which a strict format is compensated by the extreme ease with which it can be learned and used, the interactive terminal being an ordinary teletype.

In the multimedia physics course taught at the Naval Academy (Vierling and Serlimitsos) we have another one-time educational experiment. In this case the computer is an adjunct, rather than the chief mode of administration of learning materials. Eight different student sections were cycled through different instructional techniques, from books to video tapes, and including computer-assisted instruction. The computer was also used extensively in managing the instructional process and evaluating its results. Although computer usage is not central to this paper, the experience reported here is relevant to the theme of the conference. Whether one subscribes to the method employed (publication is not necessarily to be viewed as endorsement by the CCP), this experiment was a very ambitious and expensive one, and merits attention by members of the teaching profession.

The papers by Zinn and Bunderson, both psychologists, were an attempt to make science teachers aware of quite a different viewpoint: that of the learning theorist. It will be seen that the concerns expressed in these papers, such as "instructional design," preparation and dissemination of materials, and "performance objectives" are distinctly different in character from those which infuse the earlier papers in this volume. Bunderson draws the conclusion that "simulation alone is inappropriate for teaching totally new material." A certain amount of expository, well-structured instruction is necessary. Expecting the student to discover very much on his own is inherently uneconomical and may even be self-defeating. The methodology involved in Bunderson's paper, which included the concoction of an imaginary science, should broaden the reader's awareness of the problems inherent in designing tutorial materials. One point emphasized in both papers is the necessity for careful *evaluation* of the results of educational experiments. God-given "right opinion" may no longer be adequate for the justification of such experiments (especially when applying for a grant).

A Computer-Assisted and -Managed Course in Physical Sciences

ORA M. KROMHOUT, DUNCAN N. HANSEN, and GUENTER SCHWARZ

INTRODUCTION

This paper describes the use of computers in physics instruction at the Computer-Assisted Instruction Center of the Florida State University (FSU) in Tallahassee. The CAI Center, which is devoted to research in the *instructional* use of computers, utilizes an IBM 1500 Instructional System for a variety of research projects in diverse subject areas, grade levels, and roles in the learning process.

The Center began operations in September 1964, and one of its first large projects was the development and implementation of a complete course in college physics, presented by the computer and other media. It was not part of the work plan to develop a new course, but rather to present, by means of the computer and other media, an already well-established course. The project, funded largely by an Office of Education grant for the three-year period from 1966 to 1969, is significant not as a novel presentation of physics, per se, but for demonstrating the use of the computer and other media to perform both the management and teaching roles of the instructor in a complete course, given for credit. It must be emphasized that the instructor is *not* replaced but functions in his creative role by planning and modifying the course.

The course selected for implementation was Physics 107, Fundamentals of Physics, a one-quarter course with optional laboratory. (The laboratory was not included in the CAI version.) The course is designed to acquaint non-science students with physics; no mathematics beyond high school algebra is expected. Since the course can be used in partial fulfillment of basic studies requirements, a majority of FSU freshmen enroll in it; this typically results in three sections of up to 250 students each being taught during each quarter of

Computer-Assisted Instruction Center; third author is director of the Center for Research in College Instruction of Science and Mathematics and in the Department of Physics. Florida State University, Tallahassee, Florida.

the academic year and one section in the summer quarter. Shortage of staff and the size of the enrollment make it necessary to give Physics 107 as a lecture course of about 30 one-hour lectures. The organization of the material has been modeled after the PSSC (Physical Sciences Study Committee) high school course, with optics and the theory of light as a basic unifying theme. The PSSC movies are frequently used instead of demonstration experiments.

This course seemed to provide a good opportunity for a CAI presentation. Initially, computer review lessons were developed and used in conjunction with the course without making any changes in the course itself. When this proved successful, the development of the full CAI course was undertaken. The success of the review lessons and the support of the regular course instructor were instrumental in gaining University permission to give the CAI version for full credit. This permission was granted with certain conditions:

(1) the course instructor would retain final editing control of all the CAI physics materials;
(2) if, in the judgment of the course instructor, the field test should prove to be disadvantageous to the students, he would terminate the experiment at that point;
(3) qualified physics proctors had to be available during all instructional settings;
(4) the derivation of grades would be from a common set of examinations given to both the CAI and conventional students.

CAI FOR PHYSICS 107

CAI review lessons to supplement the regular Physics 107 course were begun during the academic year 1966-67. Students were offered the opportunity to come to the CAI Center for these lessons on a completely voluntary basis. For each topic covered in the class lecture, and for each of the PSSC films, a lesson consisting of about six questions was written, edited by the instructor, and programmed into an IBM 1440 computer system with eight Model 1050 typewriter terminals. Most questions were in the multiple choice format, though a few required a typed-in answer (Figure 1).

The materials were programmed to keep returning the student to the question until he answered correctly; however, he could ask for a hint if he wished. In any case the computer's response to an incorrect answer usually contained information to help him answer correctly. At the end of a topic, he could choose another topic or could sign off, taking his "hard-copy" type-out with him. A student could also choose to begin his work with a diagnostic quiz, to find out which areas he most needed to review.

In the booklet, page 26, is the calculation of kinetic energy gained by the electrons in the vacuum tube. Calculate the kinetic energy in a similar tube with the following characteristics:

Force = 2.5 x 10**-14 n/elementary charge
Distance between plates = 2.0 mm
Current = 5 x 10**-3 ampere
Current allowed to run for 1 minute

Is the kinetic energy

a. 1.55 joules
b. 94 x 10**3 joules
c. 94 joules

hint
The electric charge/sec = (5.0 x 10**-3) (6.25 x 10**18) el.ch./sec = 3.125 x 10**16 el.ch./sec

Now substitute this in the equation, along with the force, distance, and time (in seconds). Watch your powers-of-ten carefully. If you still can't do it after a good try, type hint2.

hint 2
Energy gained=

(2.5 x 10**-14 nt/el.ch.) (2.0 x 10**-3 m) (3.125 x 10**16 el.ch./sec) (60 sec) = 2.5 x 2.0 x 3.125 x 6 = 94 joules.
So type c.

don't know
Type a, b, c, or hint.

n
Call the proctor.

a
No. Perhaps you forgot to change the 1 minute to 60 seconds. Try again.

b
No. Did you remember to express 2 mm as 2 x 10**-3 meters? Try again.

c
Very good! And since this equation, which uses both the Millikan elementary charge (in the force/el.ch. factor) and the Faraday elementary charge (in the elem. ch./sec factor), correctly predicts experimental results, we conclude that these two units of charge are the same.

FIGURE 1. Typical question from the IBM 1440 review lessons with responses to both the correct and the wrong answers.

The response of the students to these materials was very good. Within a few months of the beginning of this instructional service, use of the system was close to saturation. A 1967 study of the grades of students actively participating in the CAI review showed that they averaged about 10% higher than those of students who did not. It must be kept in mind, however, that an unknown factor of selectivity might be present; students who volunteer for this review might be more interested or motivated on the average than those who did not. On the other hand, they might be those who are most in need of help. In any case, there is evidence that a review service of this type is helpful and well used. This review was offered again during the spring quarter of this year on teletypes interfaced through a PDP-8 to the 1500 system. The materials are now being coded into FOCAL for implementation on the PDP-8 *TS* System, it is hoped in time for use during the fall quarter of 1970-71.

The fully autonomous CAI version of Physics 107 was prepared in 1967 and was taken for credit by twenty-three students during the fall quarter and by another thirty-seven students during the spring quarter, 1968. (During the winter quarter we were analyzing and revising the materials.) The IBM 1500 Instructional System with 31 CRT (cathode ray tube) terminals and other audio-visual materials were used. The students attended no conventional lectures, but proceeded at their own rate by making individual appointments at the CAI Center. The language used was COURSEWRITER II, which is suitable to the primarily textual material, little mathematical manipulation being required in the questions. This language also provides for the keeping of the detailed student response records which were part of the project.

Limitations on available computer time required the selection of a sample of 30 to 40 students from the 600 or so registering for Physics 107. The University administration requested that participation be on a voluntary basis. Many more students volunteered than could be accommodated, and participants were chosen randomly from the volunteers. Rough polling showed that about 70% of the students would have been interested in volunteering. Except for their willingness to try an unconventional course, we have no reason to believe that our student sample was unrepresentative of the course population. When asked their reason for volunteering, almost half said that they were curious about computer-assisted instruction; others chose it so as to be able to progress at their own speed.

Table 1 compares the structure of the CAI course with the conventional course. For each component of the conventional course, there is a corresponding one in the CAI course. In addition, each CAI lesson includes two or three short quizzes, which are intended to focus the student's attention on the main points of the material he has just covered. Both courses used the same reading assignments, and took the same midterm and final exams; but the CAI students had to pass a diagnostic quiz, usually four to six simple

TABLE 1 Outline Comparison of CAI Course with Conventional Course

Conventional	CAI
Number of students: Up to 250 per section, three sections	23 (fall 1967) 37 (spring 1968)
Textbook reading assignments	Textbook reading assignments
	Quiz on Assigned Reading at 1500 CRT terminal before each lesson. Student must pass to go into lesson.
Lectures: 28-30 (3 per week at fixed hours) (a) spoken lecture (b) demonstrations (c) blackboard notes: equations, diagrams, key concepts, etc.	29 (self-paced) (a) audio lecture (b) 4 - minute concept film loops (c) booklet of mimeographed supplementary sheets.
	Quiz on lecture contents at 1500 CRT terminal: conceptual and quantitative questions.
PSSC Movies: About 20 during lectures	Same, self-paced.
	Quiz on ideas presented in PSSC movies at 1500 CRT terminal.
Homework: Two sets not collected or graded.	Homework: Two sets not collected or graded.
"Outside" help available: (a) Graduate students with consulting hours (b) CAI review on IBM 1440 system.	(a) Proctors at CAI Center (b) CAI review on 1440 system.
Mid-term and Final Exams: Multiple choice, determine course grade.	Same, but may be taken earlier than date scheduled in conventional course if desired.

questions, before they were allowed to progress into the main body of a lesson. The computer kept their score, and automatically told those who didn't achieve the criterion level to sign off and come back when better prepared to take the quiz again. In practice, some students looked up the answers in the text as they went through the quiz, but, in any case, they had some familiarity with the text material before listening to the lecture.

The CAI student was then directed by the computer to listen to a taped lecture. The content of these was the same as for the conventional course, but special arrangements had to be made to replace lecture demonstrations and the blackboard jottings which are at the disposal of a classroom lecturer. Silent four-minute color film cartridges were used for the demonstrations. At the appropriate place in the lecture, the student was directed to watch a particular demonstration using the Technicolor cartridge projector, which is extremely simple to operate.

About 27 of these film loops were used. Some were purchased commercially, but most were made on campus as needed. The film loops had the advantage that they gave each student a close-up, detailed view of the demonstration, and could be viewed as many times as desired. Their main disadvantage when used without an instructor was in not having a simultaneous explanation, although mimeographed film notes were provided.

A classroom instructor normally writes down equations and concepts as he lectures to a class, and we felt that some visual reinforcement was also necessary for the CAI lectures. For this purpose, a booklet of "supplementary sheets" was prepared and given to each student to follow while listening to the taped lecture. Some blank space was left on the sheets for any additional notes the student might want to make. Many students commented that this booklet was one of the most useful aids in the course, not only before and during the taped lecture, but also for review before exams.

Each lecture was followed by a quiz at the computer. Most questions were multiple choice, but some required a constructed response. The student could ask for a hint or try an answer, but he would not be moved on to the next question until he answered correctly. If he was wrong, the computer would say so and then give him some help toward finding the right answer. A record of his responses was kept for data analysis, but no "score" was kept on a given quiz, nor was he branched back to repeat material, as was the case for the diagnostic reading quiz.

Twenty of the twenty-nine lessons included a PSSC movie of 20 to 30 minutes running time. At the appropriate point in the lesson, the computer instructed the student to see film number X, and then return to the terminal for a film quiz. The logistics of showing twenty fairly long movies, as many times as there were students, with two 16-mm projectors was the biggest bottleneck in the course. A different arrangement, such as cartridged films, should be worked out if such a course were to be given regularly.

Like the conventional students, CAI students could come to the Center for computer review lessons if they wished, and some did. The only human help provided for the CAI students were the proctors at the CAI center. There was always a proctor in the room to help with the equipment, find films, and so on. A physics proctor was available at the Center, but usually not in the same room. The students rarely left the terminal room to find the physics proctor, though they would readily ask him questions if he was in the same room.

At the end of each lesson the computer gave the text reading assignment for the next lesson, after which the student signed off or went on to the next lesson as he preferred.

ANALYSIS AND RESULTS

The computer kept a continous record of each student's day-by-day performance, recording every response, whether by light pen or keyboard, and the latency or length of time in tenths of a second that he took to respond. At the end of the course, the data analysis programs were used to pull out several types of record from the basic tape of responses. First, the responses were grouped by student, generating for each student a complete record of all his responses and the latency for each response. These individual student records are of great interest to the educational psychologist.[1]

A second type of analysis, more useful to the instructor and writers, is the item analysis, Table 2. For each item, or question, the program gives the total number of "first-pass" correct and incorrect responses for the class as a whole, as well as the average latency for each type of response. For multiple choice answers, this information is given for *each* of the answer choices. Thus, if the same wrong answer is chosen by a substantial portion of the class, the author might infer that the material was not being presented properly or that the question was poorly stated.

A third type of printout, again useful to the instructor and writers, is a record of the unanticipated responses to questions requiring a typed-in answer. This helps the instructor to recognize areas in which the class or an individual student did not have a clear understanding of the subject matter, and enables the writer to add to the list of recognized answers (whether correct or incorrect). The advantage of having as many student answers as possible on the "recognized" list is that an appropriate computer response can be written for each that clears up any misunderstandings implied by the student's response.

These analyses were used as a basis for revisions in the material between the first and second presentations of the course. The need for more basic, long-range revisions was indicated by the data on first-pass responses grouped by subject area. This showed an increasing number of incorrect answers in the final sections of the course, on electricity and magnetism and modern physics. This finding was substantiated by interviews with students, who found that the course moved much more quickly in the last half, and the subject matter seemed more difficult.

Students were interviewed individually after their final examination in the spring of 1968. More than 80% said that they liked the CAI approach, would take other courses this way if the opportunity arose, and would recommend it to a friend; although several restricted this to a friend who is "good at

TABLE 2 Example of an Item Analysis Summary, Physics 107. Shows total class performance on the "first pass" for each question designated by an "Identifier" number. N is the total number of students answering the question; ID describes the type of response: C1—correct; W_n—n-th wrong alternative; UU—unrecognized wrong response. N(I) is the number of students who chose the response in ID; P.C. represents N(I)/N in percent; "Avg. Lat." is the "average latency," or average student response time in seconds; S.D. is the standard deviation of the average latency data

Identifier	N	ID	N(I)	P.C.	Avg.Lat.	S.D.	ID	N(I)	P.C.	Avg.Lat.	S.D.
20C01L	23	UU	23	100.0	6.84	9.930					
20C03C	22	C1	17	77.2	13.93	17.918	UU	1	4.5	30.90	0.006
		W1	3	13.6	14.10	8.774	W2	1	4.5	23.40	0.006
20C04C	22	C1	22	100.0	11.80	8.659					
20C05C	22	C1	11	50.0	30.43	12.826	UU	1	4.5	18.40	0.004
		W2	8	36.3	30.99	15.071	W3	2	9.0	29.40	5.400
20C06C	22	C1	18	81.8	12.28	8.213	UU	1	4.5	34.80	0.010
		W2	2	9.0	8.05	0.050	W3	1	4.5	27.40	0.003
20C07C	22	C1	15	68.1	9.48	5.184	W2	5	22.7	7.07	3.105
		W3	2	9.0	16.30	4.099					
20C08C	22	C1	18	81.8	21.91	13.999	W1	4	18.1	17.20	5.117
20C09C	22	C1	21	95.4	11.45	7.737	W1	1	4.5	6.40	0.001
20C10C	22	C1	21	95.4	21.13	21.493	W3	1	4.5	12.00	0.000
20C10R	22	UU	22	100.0	14.94	14.690					
21A01C	29	C1	28	96.5	14.38	15.499	W2	1	3.4	10.00	0.000
21A02C	29	C1	22	75.8	18.77	16.715	W1	7	24.1	34.11	21.440

digging things out for himself." The self-pacing aspect was appreciated by all, and only a third felt that the CAI approach caused them to have less personal attention. Multiple choice answers were preferred over typed-in answers by 85% of the students. Because of the speed of presentation, more than half preferred the CRT system to the typewriter terminal; however, students coming in for the review lessons preferred the typewriter, because of the "hard-copy" they could take with them.

Many students commented on the need for verbal explanation with the film-loop demonstrations and the importance of having well-trained proctors readily available to locate and trouble-shoot equipment as well as to answer questions on the lesson content. Students could not by-pass the quizzes on the computer, but they could, if they wished, skip other components of the course, such as lectures or movies. The interviews showed that about 75% of the students voluntarily by-passed some of the materials.

We were interested in learning to what extent students answered incorrectly on purpose in order to see the responses from the computer to the various wrong answers. If there were much of this, it would affect our interpretation of the number of first-pass correct answers in the data analysis. Forty percent said they did not deliberately answer wrong, forty-nine percent did it "a little" and 11% did it a lot. While this introduces some error into the data, it is not enough to invalidate them.

One of the unique features of the CAI course was that it provided for self-pacing, the only constraint being that students had to take their midterm and final exams no later than the date of the conventional class test. To see whether they actually took advantage of this opportunity, we compared their average lesson-by-lesson progress to that of the conventional class. The CAI students on the average lagged behind the regular class during the first half of the term, but then speeded up and finished, on the average, about eight days before the end of the conventional class, or 12% early. The ability to finish early made it possible for some students to take this course who could not have taken the conventional course.[2] Students usually completed one to three lessons at a sitting.

Course grades were determined entirely by performance on the midterm and final exams. CAI and conventional students took the same exams, in the same format. A comparison of grades between the two groups showed a significantly higher performance by the CAI group in the fall quarter. In the spring quarter, the CAI performance was again higher, but not enough so as to be considered significant. This indicates both the complexities of this approach as well as the advantages in terms of better performance and more favorable student reactions, an important factor in the current collegiate milieu.

COSTS

Cost reporting in this field is difficult because of problems of categorization and estimation. A foremost consideration for CAI projects is the differentiation, if possible, between course development costs and the operational costs of instruction. Table 3 presents a breakdown of the costs associated with this project. The most noticeable feature of this breakdown is the large percentage, about 95%, classified as development cost. It is true that this includes a large one-time expenditure for the development of a data management and analysis system, which was not provided by the manufacturer and which was essential to the completion of the project. Even if this is subtracted from the cost of development, however, the breakdown indicates the high cost of development in such research projects.

The instructional operational costs, when divided by the total number of student hours, came to approximately $1.78 per student-terminal hour, about half of which is hardware cost. It is estimated[1] that the total preparation cost of CAI learning materials was about $2500 per hour of instruction for the initial version and about $5000 after revision. The recent development of medium-sized, inexpensive computer systems, plus an anticipated drop in CRT terminal costs, makes the future look optimistic. The FSU-CAI Center is presently working on a 64-terminal system for which a hardware operational cost of approximately 20 cents per instructional hour is anticipated.

Another use of the computer in instruction is computer-managed instruction (CMI) which we believe holds great economic promise.[3,4,5] CMI can be defined as an automated approach to individualized instruction that may include such functions as (1) diagnostic evaluation with learning prescriptions, (2) the limited use of CAI for drill and practice or conceptual development, (3) learning simulations, (4) counseling of the student, (5) development of a scheduling system for optimally matching students with learning resources, and (6) the development of an appropriate student instructional-record system. CAI encodes the learning materials within the computer system, while CMI utilizes a combination of conventional printed and multimedia materials.

The physics course has not been presented in a CMI format, but two studies[6,7] recently completed at FSU report on CMI in a graduate-level course in "Techniques of Systematic Instruction." In this case the majority of the diagnostic evaluations and learning prescriptions take place during a computer-terminal interaction between the student and the CMI system. This allows for the inclusion of CAI techniques, when desired. within the overall approach. Secondly, it makes the student responsible for correcting any errors in the information flow both coming in and going out. Lastly, the student receives his next assignment immediately, as opposed to waiting 24

TABLE 3 Costs of Developmental and Instructional Activities for the CAI Multimedia Physics Project

I. DEVELOPMENTAL COSTS	Expenditures $	Percent of Total cost	Expenditures $	Percent of total cost
A. PROFESSIONAL MANPOWER				
1. Curriculum Preparation				
Physics Writers	30K	10		
Physics Faculty	25K	9		
Behavioral Scientists	42K	15		
Subtotal			97K	34
2. Computer Programming and Coding				
CAI Coding	12K	4		
Data Management Programming	40K	14		
Data Analysis Programming	24K	8		
Subtotal			76K	26
3. Experimental Studies Support				
Gradiate Students	24K	8		
Subtotal			24K	8
4. Administration				
Director	11.25K	4		
Secretaries	15K	5		
Subtotal			26.25K	9
B. COMPUTER AND MATERIALS COSTS				
1. Computer Time				
CAI Coding	9K	3		
Systems Programming	25K	9		
Data Analysis	9K	3		
Subtotal			43K	15
2. Media Preparation				
Films and Graphics	6K	2		
Audio	2K	1		
Subtotal			8K	3
Total Developmental Costs			274.25K	95
II. INSTRUCTIONAL OPERATIONAL COSTS				
A. MANPOWER				
Proctor	3.5K	1		
Computer Operations & Technicians	4.5K	2		
Subtotal			8K	3
B. COMPUTER AND FILM COSTS				
Computer	6K	2		
Film Rental and Audio	1.25K			
Subtotal			7.25K	2
Total Instructional Costs			15.25K	5
TOTAL PROJECT COSTS			289.5K	100

hours or more. Indications are that CMI can reduce operational computer costs by a factor of three or four as opposed to CAI.

The course itself is not presently in use. The facilities at the CAI Center are dedicated primarily to research activities, and would not be available or adequate for regular use to present a course with such a large enrollment. Also, funding for continued operation is not presently available. However, persons interested in the materials should write to Dr. Duncan Hansen, one of the authors of this paper. The course can be copied on tape for use in another IBM 1500 system, however, remote operation is not practicable. The materials would have to be recoded for any other computer system. Lecture tapes are easily copied, but we have only a few copies of the film loops at present. The Final Report, which includes typed copy of the lectures, computer materials, and student's supplementary sheets, plus other information, is available at cost from the CAI Center.

The review lessons for the course cc itinue to be used each quarter by students in the conventional course in preparation for their examinations. At present they are presented on teletype terminals, using a PDP-8 TSS system and the FOCAL language. It is hoped that as soon as our DEC tape arrives, arrangements can be made to make copies that could be used on other PDP-8 TSS systems.

REFERENCES

1. D. N. Hansen, W. Dick, and H. T. Lippert, *Research and Implementation of Collegiate Instruction of Physics via Computer-Assisted Instruction.* Final Report, Computer-Assisted Instruction Center, Florida State University, Tallahassee, November 1968.
2. O. M. Kromhout, S. Edwards, and G. Schwarz, "A Computer-Guided, General Education Physics Course," *Am. J. Phys. 37,* 995, 1969.
3. J. C. Flanagan, "Project PLAN: A Program of Individualized Planning and Individualized Instruction." Paper presented at Project ARISTOTLE Symposium, Washington, D.C., December 1967.
4. I. E. Coulson, *Progress Report for the Instructional Management System,* System Development Corporation, Santa Monica, California, May 10, 1968.
5. E. N. O'Dierno *et al. Phase I Report on a Research Study to Develop a System for Individualizing and Optimizing Learning Through Computer Management of the Educational Process.* New York Institute of Technology, 1968.
6. N. K. Hagerty, Development and Inplementation of A Computer Managed Instruction System in Graduate Training. Unpublished doctoral dissertation, Florida State University, 1970.
7. P.D. Gallagher, An Investigation of Instructional Treatments and Learner Characteristics in a Computer-Managed Instruction Course. Unpublished doctoral dissertation, Florida State University, 1970.

The PLATO System and Science Education

DONALD L. BITZER, RICHARD W. BLOMME,
BRUCE A. SHERWOOD, and PAUL TENCZAR

The PLATO III computer-based education system (Figure 1) offers students and authors quick response, full graphical display capabilities, and a great deal of number-crunching computer power. The simplicity and flexibility of the TUTOR language, coupled with the ability to switch quickly between writing and testing a lesson, have made the full capabilities of the system easily accessible to users. This environment has given birth to lesson material in a wide range of subject areas, including undergraduate science. Examples are given in this paper of diverse educational strategies in science and engineering education to illustrate these points. The basic structure of the TUTOR language is discussed, and some technical details are given to explain how the PLATO III system works. In closing there is a brief discussion of the large-scale PLATO IV system now being developed.

The examples given below are necessarily brief sketches which do not include the details of the complete lessons. Further information about particular lessons may be obtained from their authors, whose names and departmental affiliations are given in parentheses. Unless otherwise specified, they are at the University of Illinois, Urbana. Note that these authors are *users* of the system, not computer programmers, and span all academic ranks. The examples are grouped loosely by subject area and do not include a number of important PLATO activities.[1]

LIFE SCIENCES

Computer-generated fruit flies are manipulated by students in a Genetics laboratory (Prof. David C. Eades, School of Life Sciences): starting with

Computer-Based Education Research Laboratory, University of Illinois, Urbana, Illinois.
Work supported in part by the National Science Foundation under Grant NSF GJ 81; in part by the Advanced Research Projects Agency under Grant ONR Nonr 3985 (08); in part by Project Grant NPG-188 under the Nurse Training Act of 1964, Division of Nursing, Public Health Service, U.S. Department of Health, Education, and Welfare; and in part by the State of Illinois.

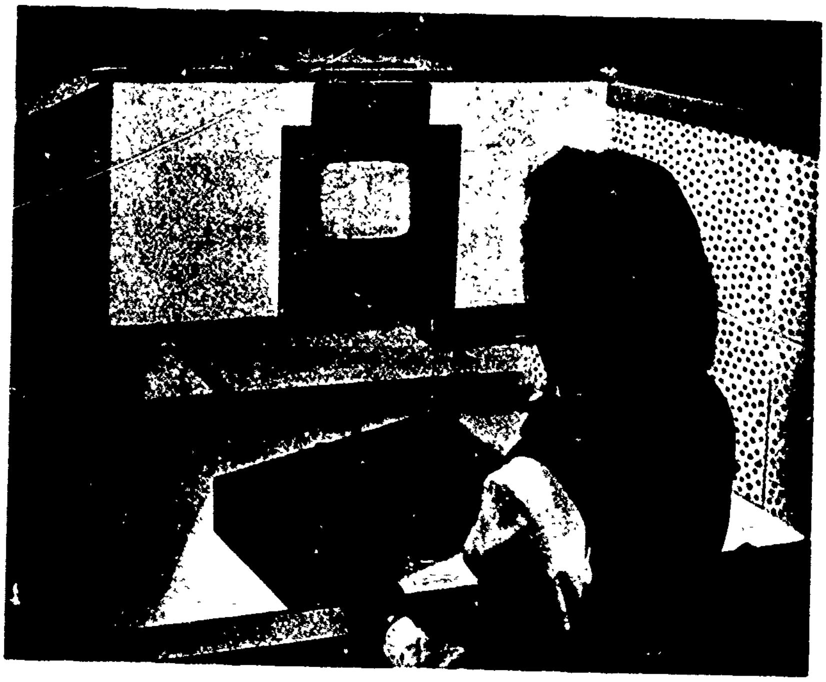
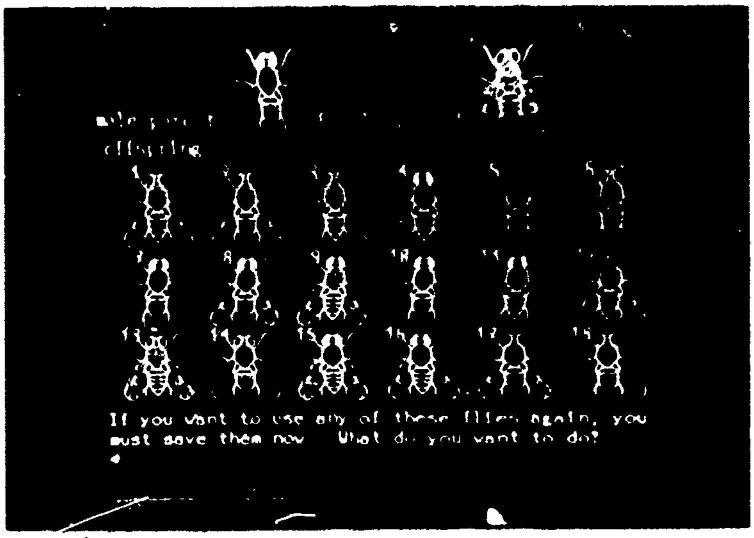


FIGURE 1 PLATO carrel.

normal and mutant stocks, the student can mate flies of his choice to produce a screenful of computer-generated offspring [Figure 2(a)]. He can save any of these offspring for future matings. Through matings and offspring analysis, the student obtains information to elucidate the hereditary mechanism involved in particular mutants. The student always returns to the basic question: "What would you like to do next?" which forces him to design his problem-solving procedure.

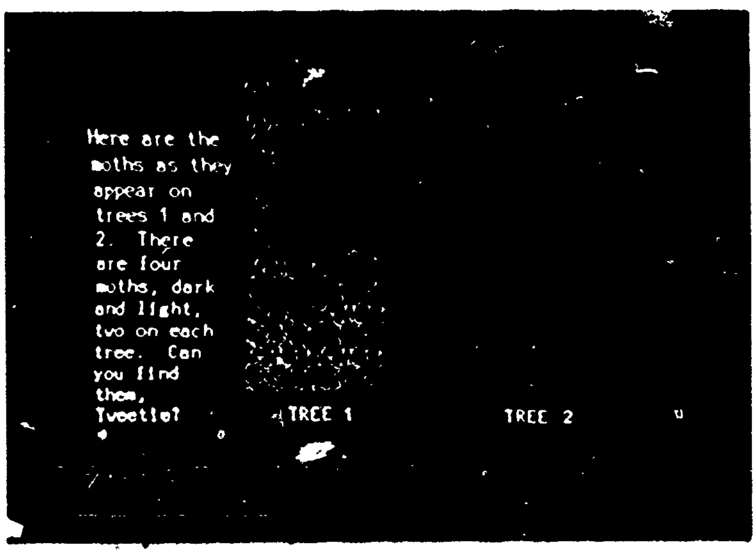
A lesson on natural selection (Prof. David C. Eades; Gary W. Hyatt, Zoology) allows the student to play the role of a bird searching for food. The "food" consists of light and dark moths displayed at random positions on light and dark trees [see Figure 2(b)]. The student finds and "eats" most dark moths on light trees, but finds few of the light moths on light trees. The process is continued through several simulated generations of moths to demonstrate the change in gene frequencies. The student thus gains insight into natural selection as a force in evolution.

In a maternity nursing course (Maryann Bitzer and Martha Boudreaux, Mercy School of Nursing, Urbana, Illinois; Mrs. Elisabeth Lyman, Computer-

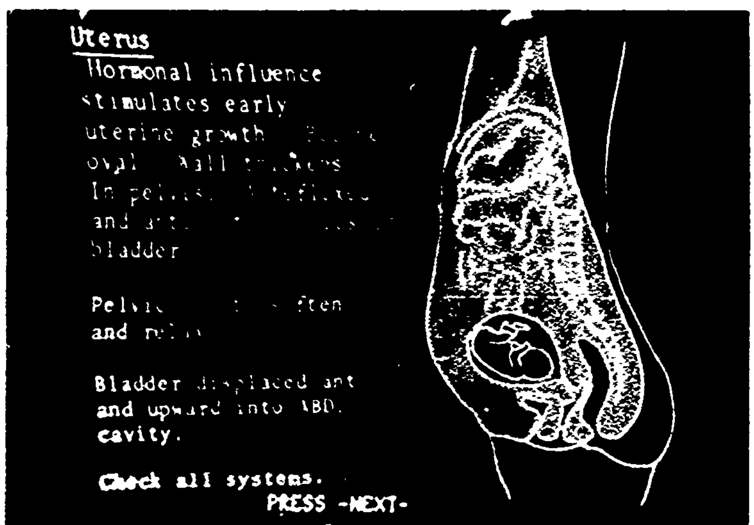


(a)

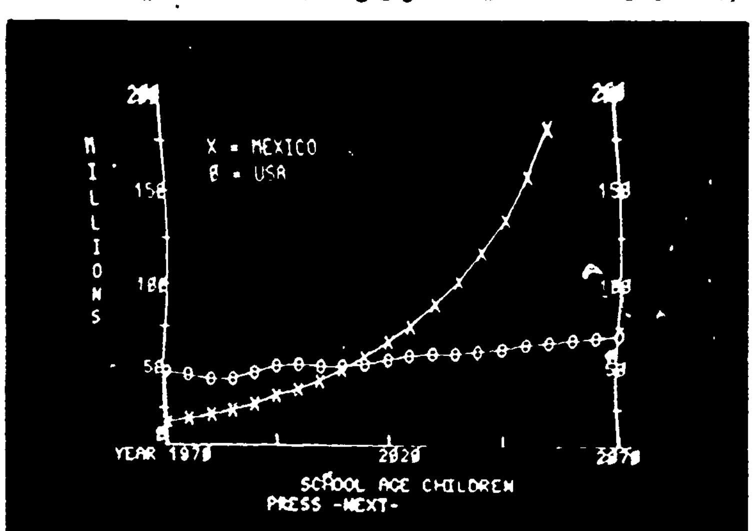
FIGURE 2 Photographs of screen from PLATO lessons in Life Sciences: (a) fruit-fly genetics; (b) moth evolution; (c) maternity nursing; (d) demography.



(b)



(c)



(d)

based Education Research Laboratory) student-directed inquiry is stressed: confronted with hypothetical patients and nursing-care problems, the students must obtain and select the information necessary to solve problems or answer questions. For instance, in determining the underlying causes of minor discomforts of pregnancy, the student can investigate the anatomic and physiologic changes that occur in various parts of the body, e.g., the uterus, during pregnancy [see Figure 2(c)]. The information obtained provides the basis for determining proper nursing strategies to be used in relieving the discomfort. The student is forced to think, investigate and experiment to gather data, categorize information, and test hypotheses, thus developing or reinforcing critical thinking skills.

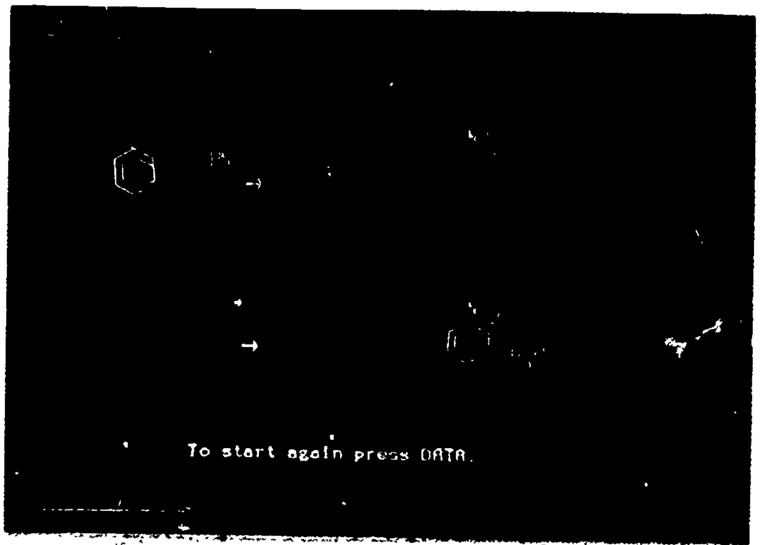
A general-purpose demography program has been developed (Paul Handler, Physics; Judith Sherwood) which allows a student to make population projections for many countries. Figure 2(d) shows a graph comparing the numbers of school-age children in the United States and in Mexico for the next one hundred years assuming present birth and mortality rates. Note the "population waves" in the United States data, caused by the depression dip and postwar boom in births. The student can change basic assumptions, such as the average number of children per family, and see the effects of these changes on the population projections. The student can plot many other demographic variables including a bar-graph of the projected age-group distribution.

CHEMISTRY

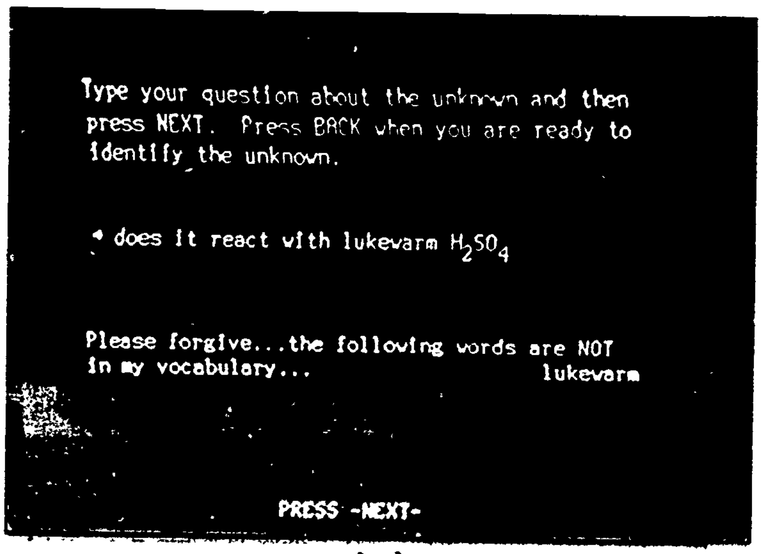
Students can perform multiple-step organic syntheses for electrophilic aromatic substitution reactions by indicating the reagent for each step in the synthesis (Prof. Stanley Smith, Chemistry). The computer displays the actual product of each specified reaction, checking for compatibility of reagents with functional groups present, reactivity levels, and proper orientation [Figure 3(a)]. Help on the introduction of any functional group is provided. Students are free to choose any sequence of reactions they wish in attempting to synthesize the desired end product.

Experience in rapid identification of unknown organic compounds is provided by a lesson (Prof. Stanley Smith) in which the student simply types questions about the compound [Figure 3(b)]. The computer provides essentially instantaneous answers, such as giving the melting point or showing the nmr spectrum of the unknown compound. The vocabulary of the program is adequate to answer almost all the experimentally useful questions about the compound under investigation.

Another lesson (Robert Grandey, Chemistry) permits students to generate their own problems or work problems provided in the lesson [Figure 3(c)]. If the student supplies a chemical equation, he can include formulas for molecules, elements, charged ions, and electrons. The computer determines if the equation is balanced both by mass and charge and provides appropriate error messages. Problems of the following types are available:



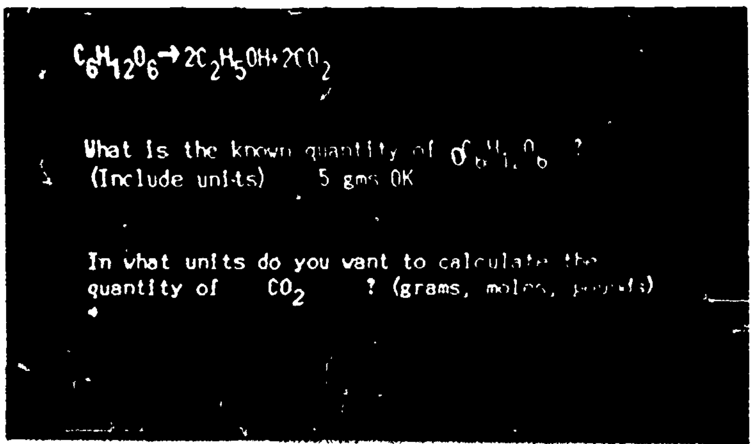
(a)



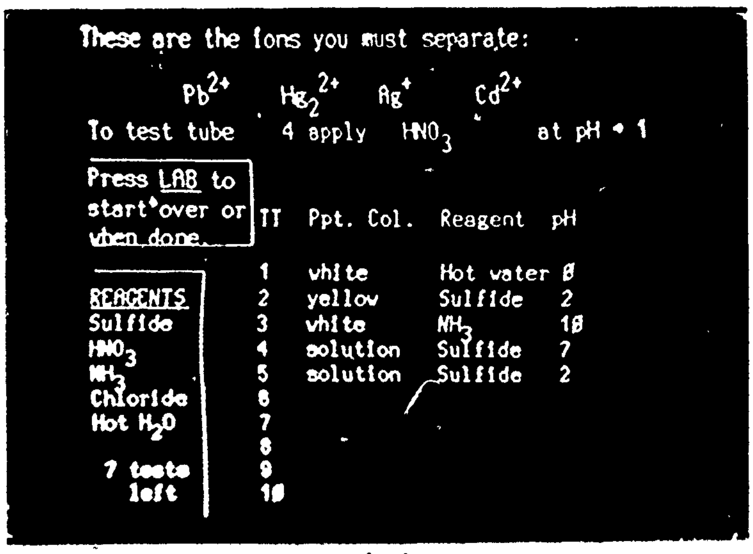
(b)

FIGURE 3 PLATO lessons in Chemistry: (a) organic synthesis; (b) organic qualitative analysis; (c) chemical equations; (d) inorganic qualitative analysis.

(1) determining the chemical formula from the composition by weight;
(2) calculating the percentage composition from the known chemical formula;
(3) determining quantitative relations in chemical equations.



(c)



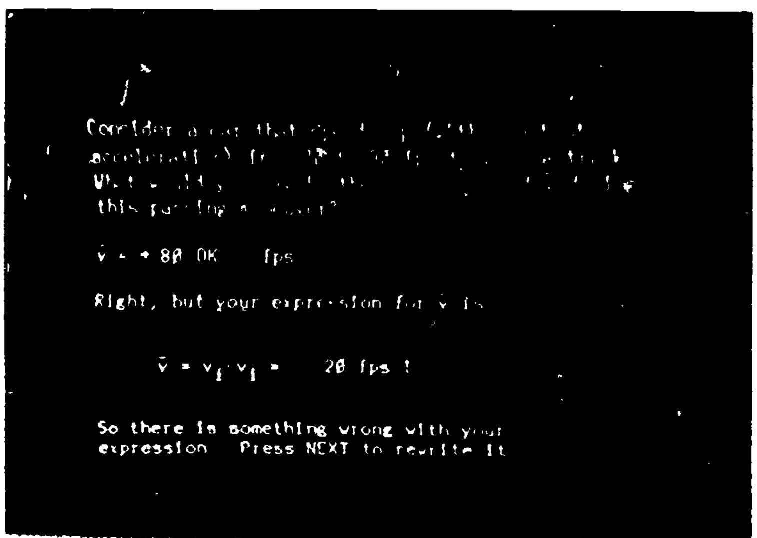
(d)

The correct answer and specific step-by-step help relevant to the particular problem are generated by the computer. At all times the student may use the computer as a calculator.

Students using another program (Larry Francis, Chemistry and Chemical Engineering) can work with many more inorganic ion samples than would be possible if they worked only in a chemistry lab. A student adds chemicals from a computer-supplied list to a test tube containing a randomly generated set of ions. The precipitates and solutes are poured into different test tubes. There are numerous ways to solve each problem, and a student is free to try any method he wishes. The computer simulates the laboratory results and reports them to the student [Figure 3(d)]. Options allow a student to gather additional information about his problem in the way such information is generally available in the lab.

PHYSICS

Students are led to derive kinematics relationships for constant acceleration (Prof. Bruce Sherwood, Physics): the remedial problem at the top of Figure 4(a) appears if a student gives an incorrect formula for the average velocity.

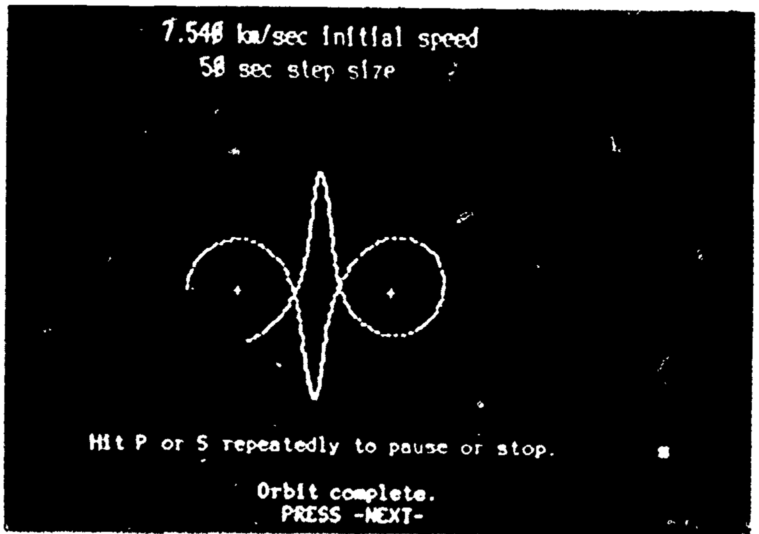


(a)

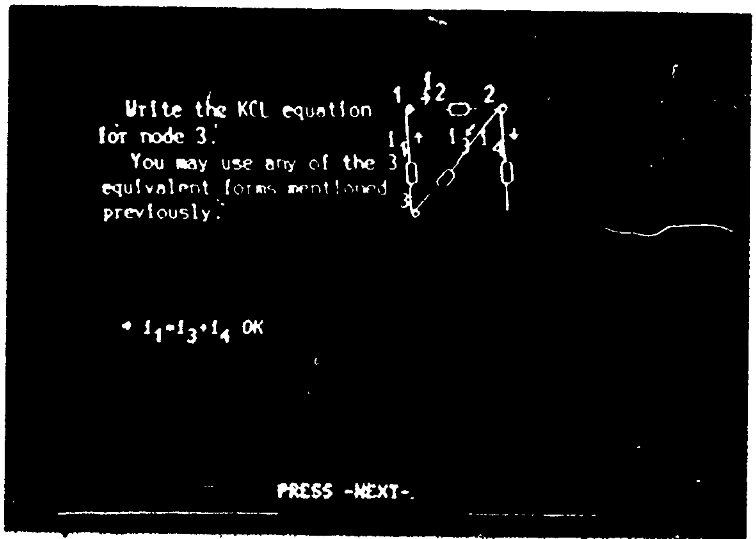
FIGURE 4 PLATO lessons in Physics and Engineering: (a) kinematics; (b) gravitational orbits; (c) quantum statistics; (d) electrical networks; (e) beam loading; (f) statics.

After answering that $\overline{v} = 80$ fps, the student is shown the inconsistency of his own formula. [The correct expression is $\overline{v} = (v_i + v_f)/2$; where v_i and v_f are the initial and final velocities.] His expression is evaluated for particular values and checked numerically, so equivalent expressions are allowed, such as $\overline{v} = v_i + (v_f - v_i)/2$. In the suceeding steps in the derivation the student eliminates v_f from the expression for $\overline{v}$, then substitutes into the definition $\Delta x = \overline{v}\Delta t$ to derive $x_f = x_i + v_i\Delta t + \frac{1}{2}a(\Delta t)^2$. This sequence is part of a lesson on kinematics which includes tutorial and calculational aspects.

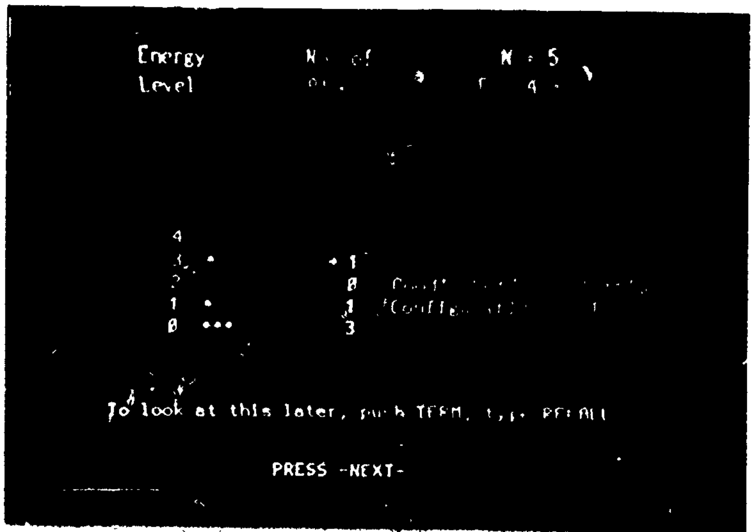
Figure 4(b) shows a very peculiar but correct orbit of a satellite about two stationary "earths" whose centers are marked by the + signs (Prof. Sherwood). The satellite was launched with a speed of 7.54 km/sec from the surface of the left "earth." The animated orbit-plotting was stopped by the student, who can now try a different initial speed. The purpose of this simulation is not only to demonstrate motion in an unusual force-field but also to show to the student the power and limitations of simple numerical integration techniques. The student can specify the integration step size to be used in the calculation and determine his own compromise between speed and accuracy.



(b)



(c)



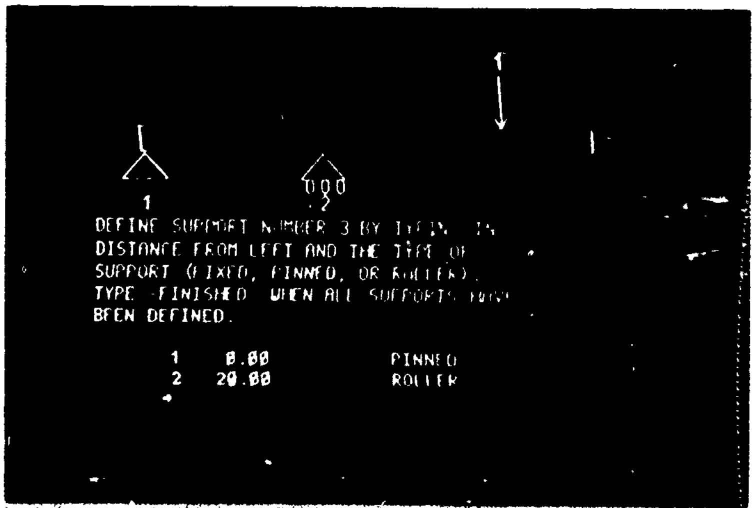
(d)

A lesson on quantum statistical mechanics is being developed (Prof. Donald Shirer, Physics, Valparaiso University) in which the student chooses *N*, the number of atoms in his system, and *E*, the total energy of the system. Figure 4(c) shows the problem the student now faces: he must specify how may atoms to place on each energy level, consistent with the chosen N and E. The student is led to discover all allowable configurations and their relative probabilities. This leads to a quantitative discussion of the equilibrium distribution as being represented by the most probable configuration.

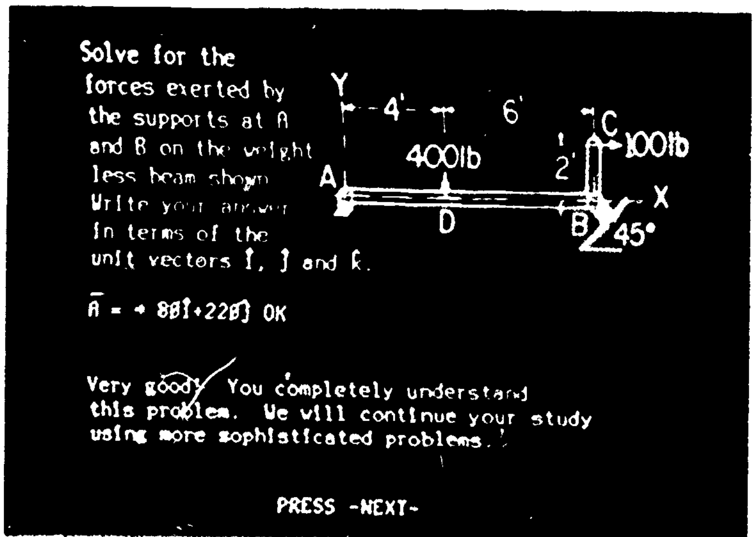
ENGINEERING

A lesson on Kirchhoff's laws (Roger Grossel, graduate student, Electrical Engineering) is part of an introductory electrical networks course: for the current law, a network with branch current symbols is displayed [Figure 4(d)]. The student uses the keyboard to assign his own reference directions. He is then asked to type the current law equation for various selected nodes. The computer algorithmically generates the correct answer from the network topology and the student-assigned reference directions. Depending on both the student's answer and student-available options the student is branched to remedial work, the same network, a more difficult network, or to new material. A similar procedure is followed with the voltage law.

In a Civil Engineering lesson (Doug Nyman and Prof. S. J. Fenves, Civil Engineering) a student may solve problems chosen from a problem library or he may create his own problem by specifying loading and support conditions as shown in Figure 4(e). Instructors may define problems in the same way as the student, test them, and then add them to the problem library by using a SAVE option and a password. In this way new problems can be generated by instructors who do no programming on the PLATO system.



(e)



(f)

Figure 4(f) shows a typical problem from a course in statics (Prof. T. M. Elsesser, Theoretical and Applied Mechanics) and illustrates the use of photographic material in displaying a complex structure to the student.

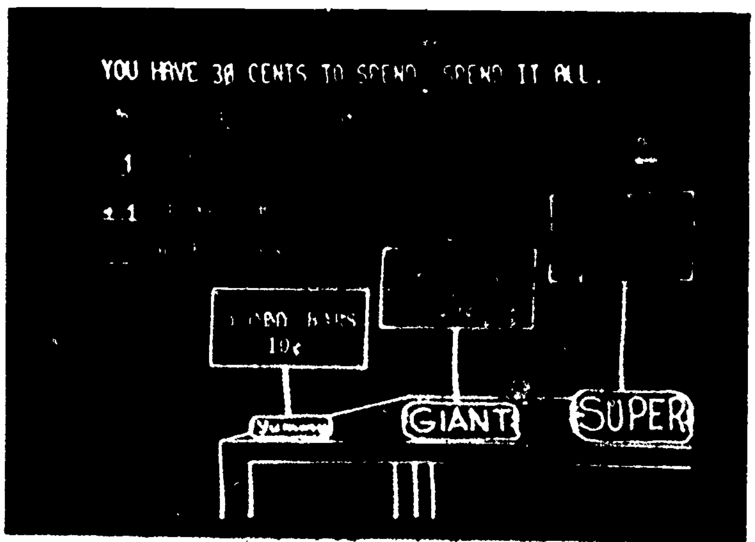
OTHER

Here are a few examples from areas outside undergraduate science and engineering which illustrate other kinds of strategies and capabilities.

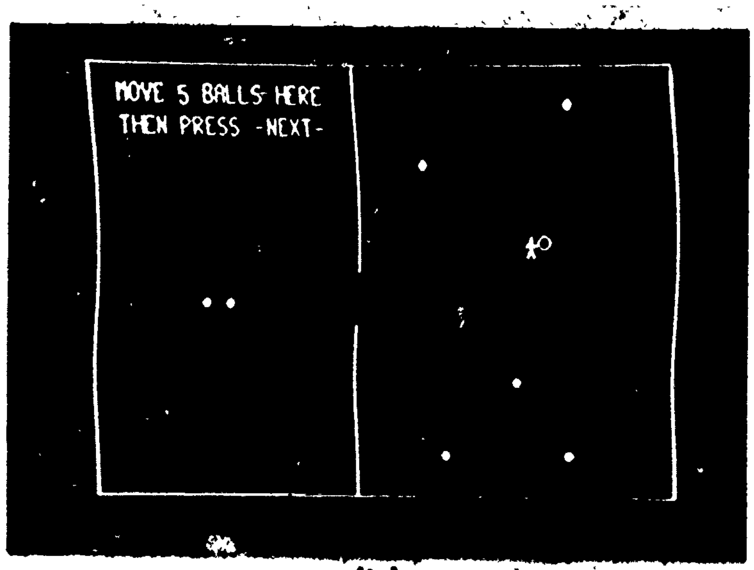
A lesson for second graders (Esther Steinberg) "gives" the student a specified amount of money and tells him to spend it all on some candy bars. A different price is marked on each kind of candy. See Figure 5(a). The student may "buy" as many of each kind of bar as he wants as long as he spends exactly the specified amount of money. Since the solution is not unique, he must do some thinking and make practical application of his skill in addition and subtraction.

In a lesson being developed (Paul Tenczar) to teach programming to grammar school children, second graders readily learn to "walk" a man around the screen to do work. In Figure 5(b) the student is well on his way to solving the problem using keys that move the man, pick up a ball, and put down the ball. Other exercises include solving a maze, finding a path home through an alligator-infested swamp, and giving a list of directions for the man to follow (i.e., programming).

Another technique (Prof. Keith Myers, French) is to display a picture of

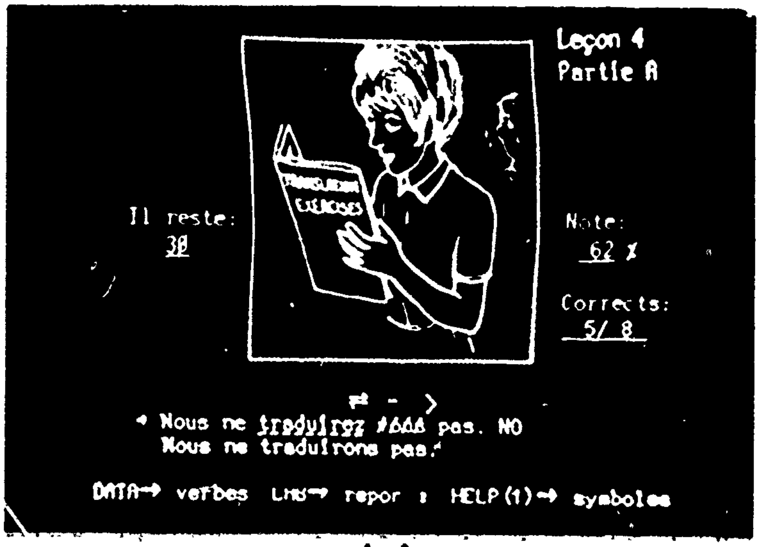


(a)

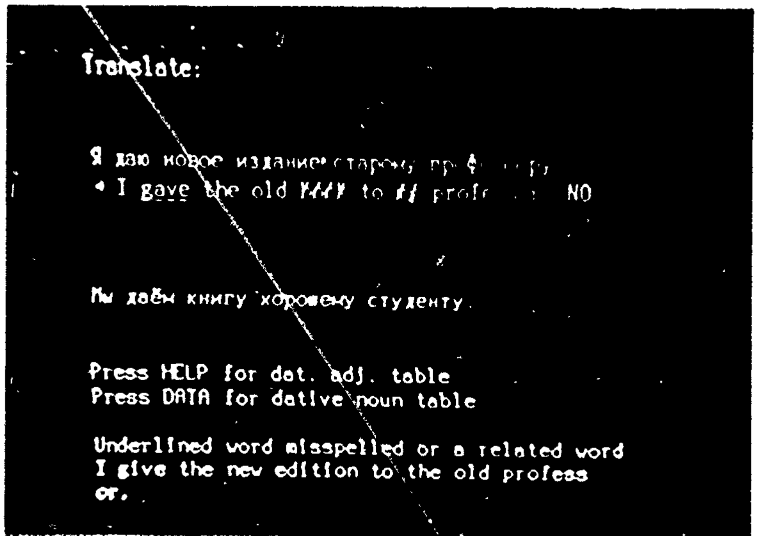


(b)

FIGURE 5 Examples of other PLATO capabilities, drawn from elementary education: (a) arithmetic; (b) simple programming; (c) French; (d) Russian.



(c)



(d)

some activity and ask the student to describe the action in French. The student is further aided by non-verbal cues just beneath the picture. In this way the entire presentation can be in French with no English instructions. In Figure 5(c) the student has answered incorrectly, and the computer has replied not merely by saying "NO" but by crossing out wrong words and underlining misspelled or nearly correct words. This technique is called "sentence judging" and is a standard feature of the TUTOR language.

Figure 5(d) is taken from a Russian course (Connie Curtin, University High School). Here the student is practicing translation from Russian into English, and his answer is handled by "sentence judging." In this particular case, upon making an error the student has also been given an acceptable translation at the bottom of the page. When the student is asked to answer in Russian he uses the standard keyboard, but Russian characters appear on his screen as he types.

THE TUTOR LANGUAGE

All the lesson material just discussed was created using the TUTOR language, a language especially designed to simplify the task of writing computer-based educational material for a graphical display terminal. It is impossible to give here an adequate description of TUTOR,[2] but a simple illustration may nevertheless be useful.

The example chosen is a multiplication drill with the problems generated randomly by the computer. In addition to checking for the correct numerical answer, this lesson segment also checks to see whether the student is adding rather than multiplying or whether a wrong answer is within ten percent of the correct answer. After working seven problems the student advances to a division drill.

```
UNIT      INITIAL
ZERO      I10              Initialize problem counter.
JUMP      MDRILL

UNIT      MDRILL
RANDU     I1,20            Randomly generate 1< I1<20
RANDU     I2,10                                1<I2<10
CALC      I3=I1xI2         Calculate product (I3) and sum (I4)
          I4=I1+I2

WHERE     810              Display at 8th line, 10th column.
SHOW      I1,3             Use 3 spaces to show contents of I1.
WRITE     x

SHOW      I2,3
WRITE     =                For example, the student sees "11x9="

ARROW     822              Display an arrow to indicate a response
                           is required; separate display generation
                           commands from response judging commands.
```

```
ANSV     I3,0                   Check for perfect answer.
ADD1     I10                    If perfect, increment counter and get
NEXT     I10;7,MDRILL,DDRILL    another mult. prob. or go to DDRILL if
                                7 problems done.

ANSV     I4,0                   Check for sum answer.
WRITE    You're adding!
JUDGE    NO

ANSV     I3,10%                 Check for close answers.
WRITE    You're close!
JUDGE    NO

WRONG                           Anything else.
WRITE    You're way off.

UNIT     DDRILL                 Division drill.
.
.
.
.
```

While this simple example does not demonstrate the full capabilities of TUTOR, it does illustrate some basic features of the language. In unit MDRILL, the student must correctly answer before he can move on to the next problem; when he erases a wrong answer the comment he receives is automatically erased, but the problem remains on the screen. The ARROW command separates those commands that set up the original display of the problem from the response-handling commands. The student's response will appear on the screen just after the position specified by the ARROW command.

Suppose the problem is "11 X 9" and the student types "95". Then neither the "ANSV 13,0" nor the "ANSV 14,0" specification is met. The commands following these ANSV commands are ignored. Because the answer is within ten percent of the correct answer, the commands following "ANSV I3,10%" are performed to tell the student "You're close!" and to judge the answer "NO". The student must now erase his answer and try again. On the other hand, after a correct answer the lesson will increment counter I10 and › to the next problem. Note that TUTOR often executes commands in a contingent rather than linear order. The "NEXT I10-7,DDRILL, MDRILL" is an example of branching which depends on the value of the computed quantity, "I10-7".

TUTOR authors can use up to 139 variables, labeled 1 through 139. A single letter preceding the number indicates the format of the varable: I for integer, F for floating point, and A for alphanumeric. In this multiplication drill, integer variables are used to generate a problem, to calculate expected answers, and to count how many problems have been completed.

While this drill emphasized one particular kind of numerical judging, using the ANSV command, there are many other answer-judging commands and techniques available in TUTOR for handling numerical and non-numerical student responses.

TECHNICAL DETAILS

The PLATO III system is oriented around a graphic display terminal. It is *X*/*Y*-addressable and has its own memory to free the computer from the burden of refreshing the display. The PLATO III system has one storage tube for every terminal. Computer-generated material—alphanumeric text, lines, and special characters—is first written on the storage tube in a point-by-point fashion. Then, the material on the storage tube is mixed with photographic material (if desired) and presented on a standard TV screen.

The system is oriented around core memory rather than disk memory. No disk accesses are made for students while they are studying a lesson: the lesson material and the pointers and counters pertaining to the individual student reside in core. The elimination of disk accesses during lesson presentation results in much faster response than is typical in multiterminal systems.

The PLATO III computer is a CDC 1604 with 32000 (32K) 48-bit words. Half of core (16K) is occupied by the TUTOR executor and other system routines. Of the other 16K, 10K is reserved for lesson material and 6K is divided into 20 "student banks" of 300 words each. These 20 student banks contain the complete status of the individual students signed on at each of the 20 terminals. A student bank contains the 139 TUTOR variables, answer storage, pointers marking the student's place in his particular lesson, and other systems variables. Typically a student works through approximately 3K of lesson material in an hour. During a Russian class the students might be spread over 7K of lesson material, so that 3K of lesson space would be available to chemists or political scientists for debugging their own lessons.

The integration of the three major blocks of core (executor, lesson material, and student banks) can best be described by following the sequence of operations initiated by a student pushing a key on his keyset. The keypush interrupts the computer, which pauses just long enough to add the key and its accompanying terminal identification number to a list. Eventually the computer services this request, at which point the terminal identification number is used to reference the correct student bank. Often the only operation required is to write the letter corresponding to the keypush on the student's display screen. It is important, however, to point out that any keypush may generate an "unusual" letter (Russian, for example), an accent mark, etc., or initiate a complex display. For this reason it is essential that all keypushes pass through the main computer to be processed.

Which lesson the student is studying, and where he is within the lesson, are specified by words of the student bank. These are used when necessary to reference the lesson material, which is examined by the executor to determine what to do for this student. As outlined in the discussion of the simple multiplication drill, the TUTOR executor judges the student's response and replies to him by performing those commands which follow the ARROW command. The data pertinent to this particular student are found in his student bank. If the computer's reply consists of a small amount of text or graphics to be added to the display now on the screen, the reply appears almost instantaneously. If a completely new display is to be generated, the speed of the reply is determined by the characteristics of the storage cathode-ray tube: a full screen of text and graphics takes one to two seconds. The average reply time (including short and long replies) is between one- and two-tenths of a second.

This summarizes the major steps taken in processing a student's keypush. No disk accesses are required while the student studies his lesson. Both his lesson and his particular variables are in core memory at all times. (Student sign-on and sign-off do however generate disk accesses to get and to file away essential portions of his student bank.)

At any time a teacher can type a code word and put his terminal into "author mode." Authors are permitted disk accesses to edit alphanumeric text and to compile that text into the lesson area of core memory. These operations do not conflict with the simultaneous use of other terminals by other students and authors. (However, a compilation, which typically takes 15 seconds, locks out compilations by other authors during that time. This does not affect student use and text editing by other authors.)

All lesson material resides as blocks of text on disks. When an author edits his lesson he specifies the block he wants to work on. A block consists of approximately 50 lines of text. It is read from the disk into the author's "student bank," where the text is available for editing without further disk references. The first fifteen lines of lesson text in the selected block are displayed. The author may move forward in the block to display later lines. He may insert, delete, replace or copy lines or portions of lines. He can save a group of lines and deposit them elsewhere, even in other blocks. Within a line he can insert, delete, or replace characters. When the author is satisfied with the newly edited form of the block, he returns the block to the disk.

To test a lesson the author specifies which lesson or lessons to compile. Within about 15 seconds the alphanumeric text is compiled from the disk into the lesson area of core memory. Any errors discovered by the compiler are displayed. (After processing each line of text the computer services any new student keypushes.) The author can then work through his lesson as a student. He may encounter things he doesn't like, or he may get an error message. At any time he can return to author mode, delete the compiled version from memory, modify the lesson text, and test it again.

The simplicity and speed of these author mode operations coupled with the power of the TUTOR language make the system attractive to teachers from many subject areas. Many of the successful authors using PLATO III had little or no previous computer experience. Authors need no middlemen; this greatly enhances acceptance and efficiency and encourages continual lesson improvement. The wide range of author interests and disciplines has produced continual evolution of TUTOR under the pressure of diverse needs. This evolution has acted at both ends of the spectrum, making simple lessons even easier to write and making possible more complex simulations and teaching strategies.

THE PLATO IV SYSTEM

We have discussed the hardware and software features of the PLATO III system which support science and engineering curriculum development. In closing it is appropriate to look briefly at the large PLATO IV system now being developed.

The PLATO IV design includes a large central computer system of the CDC 6000 series to handle simultaneously four thousand remote graphics terminals at projected costs of 35¢ to 50¢ per terminal hour. The capabilities of the remote terminal exceed those of the PLATO III terminal, through use of the plasma display panel, whose inherent digitally addressable memory makes possible high-quality graphics with low data-transmission rates. The panel is flat and transparent, which permits the superposition of computer-generated graphics with back-projected film images in full color. The PLATO IV terminal contains a character generator and a line generator, as well as other built-in functions. (For details on the overall PLATO IV system design, see Bitzer and Skaperdas.[3] For a discussion of some of the implications, see Alpert and Bitzer.)[4]

Lesson material and TUTOR software already created on PLATO III will be operable on PLATO IV. The student banks and lesson material will reside in an auxiliary core memory (Control Data Corporation "Extended Core Storage" or ECS). When needed, portions or all of a particular student bank and related lesson material can be rapidly transferred from ECS into the computer's central memory. It is planned that the PLATO IV system will have two million 60-bit words of ECS, an amount which should be sufficient to hold 4000 student banks plus 250 hours of lesson material. Thus, at any one time, students would have access to about 250 lessons selected as needed from a large disk lesson library.

REFERENCES

1. E. R. Lyman, *A Descriptive List of PLATO Programs, 1960-1970*, CERL Report X-2, Computer-Based Education Research Laboratory, University of Illinois, Urbana, 2nd. ed., May 1970.
2. R. A. Avner and P. Tenczar, *The TUTOR Manual*, CERL Report X-4, Computer-Based Education Research Laboratory, University of Illinois, Urbana, January 1969.
3. D. L. Bitzer and D. Skaperdas, *PLATO IV—An Economically Viable Large-Scale Computer-Based Education System*, CERL Report X-5, Computer-Based Education Research Laboratory, University of Illinois, Urbana, December 1968.
4. D. Alpert and D. L. Bitzer, "Advances in Computer-Based Education," *Science, 167*, 1582 (1970).

Templates for Conversational Programs in Physics

NOAH SHERMAN

One of the chief difficulties in the development of computer-based conversations is the amount of time required to program them. Estimates of the time required to produce enough conversational material for a one-hour physics lesson range from 10 hours to as much as 100 hours, depending upon the optimism of the estimator and the intricacy of the programs. In any case, the time demanded of the lesson designer of computer-based instructional materials is considerable.

Most of the time is devoured by two main tasks: one is *designing the pedagogical logic*, i.e., the sequence of exchanges that conversations may follow. This requires the lesson designer to construct a series of statements or questions that the computer will present to the student which require student responses. These will be input data to the conversational program. The lesson designer must anticipate a wide variety of possible responses to the statements and questions presented to the student by the computer. For each class of responses that he anticipates the student may type (on the terminal), he must make decisions on how to compare the equivalent inputs and on what the computer should do following that class of response. The possible branches that conversational exchanges in physics may take can be enormous and serve to discourage many potential contributors to computer-based conversation.

The second task in developing such conversations is that of *programming the conversational network* so that a given computer can play its role in the dialogue. The time spent on the first task depends on the individual lesson designer and the very broad range of skills that he can bring to perform it; it is the purpose of this paper to offer a means of reducing the time spent on the second task by the use of programmed "templates."

The word *template*, here, means a sequence of instructions to the computer for processing a particular network of conversational exchanges and connotes

Department of Physics, University of Michigan, Ann Arbor, Michigan, and Lawrence Hall of Science, University of California, Berkeley, California. Work supported in part by the National Science Foundation.

a particular pattern for the presentation of questions, answers and other basic elements of a tutorial conversation. It may be compared in spirit to a standard "subroutine" or "procedure" in computational programs, but the "arguments" of a template are the basic elements of the conversation, itself. The pattern determines only the logic of the conversational exchanges not the content of the conversations themselves.

A fairly wide variety of conversational exchanges can be connected together to produce a computer-based conversation using *only one template*, as will be shown in this paper. If a lesson designer can work within the branching opportunities provided by a given template, then each sequence of exchanges that can be arranged in accordance with this pattern can be programmed "once and for all," using the single template. The textual contents of such conversations are then treated as "data" for the template program to process. However, the most important consideration is that *a lesson designer does not need to know any programming language at all* in order to design conversational programs in physics.

Figure 1 shows the logic of a template for a pattern of conversational exchanges in the form of a flow diagram. J is an index which numbers the questions under consideration. In the upper left of the figure QJ denotes the text of the Jth question presented by the computer to the student seated at a terminal. The flow line to the right out of the box labeled QJ leads to the student's input which can be anything the student chooses it to be. To the right of the student's input the flow diagram shows a series of boxes. The label M in each box means "match." The MGJ box means that the student's input is compared with each of anticipated responses in the class of "good" responses for the Jth question. (The "OK" in the box symbolizes a match between student's input and a particular "good" response.)

Following MGJ :. the series of "matching" boxes are the boxes MBAJ, MBBJ, etc. Each of these corresponds to a class of responses considered inadequate for the Jth question: "bad" match of type A, "bad" match of type B, "bad" match of type C and so on. The number of such alternative classes that can be anticipated can be chosen freely by the lesson designer, and the template can accommodate as many as he chooses. On the extreme right of the matching boxes is a box labeled NO MATCH. It is at this location on the flow diagram that the program arrives in the event that the student's input is not anticipated by any member of the classes of anticipated responses.

Now follow the flow line out of MGJ, which indicates what the template does when the student's input matches an anticipated good response. In this particular template, the program checks to see whether the counter GX which records the number of successive good responses to previous questions is greater than zero or not. If it is, the flow goes out to the left following the YES line. TGX: labels the text of the comment made indicating how many

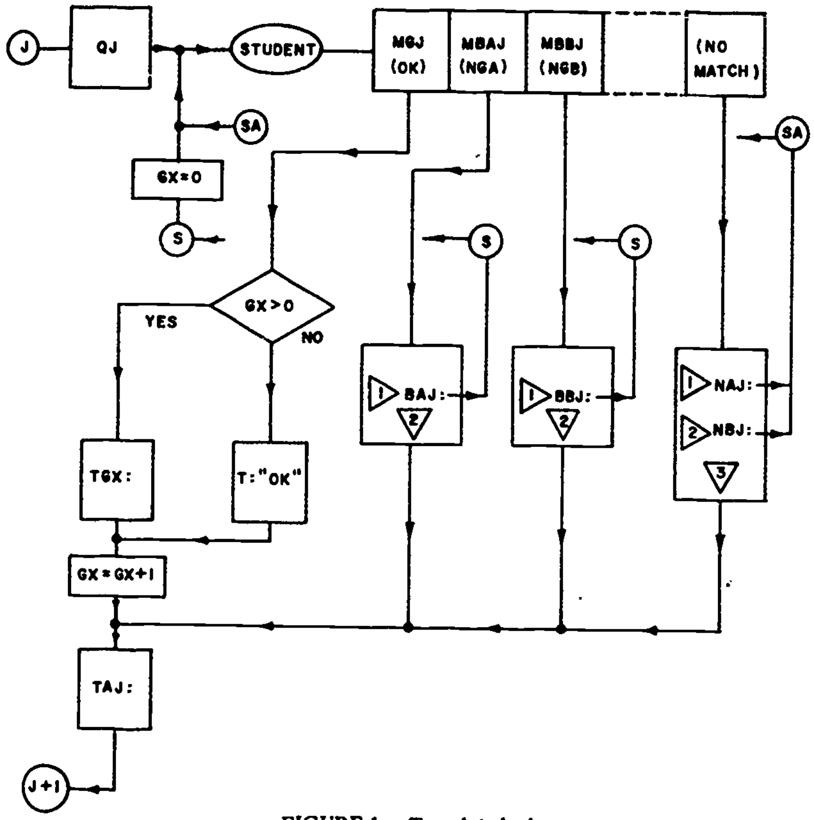


FIGURE 1 Template logic.

good responses he has made, like "very good," "right again," "excellent," and so on. Then the counter GX is increased by one, and an acceptable answer to the Jth question is presented. The text of that answer is indicated in the box labeled TAJ; and the program exits to the next question J + 1.

If the number of previous "good" responses is zero, then the NO line out of the GX > 0 diamond is followed and the first "good" comment, "OK" is made prior to presenting an acceptable answer to the question. If the student's response has been anticipated, but is considered to be less than satisfactory for proceeding to the next question, then we emerge from one of the MBAJ, MBBJ, . . . , corresponding to the class of "bad" responses matching his input. The first time this occurs a comment is presented to the student indicating the inadequacy, the counter GX is reset to zero, and the flow returns the program to accept another input from the student. This path is indicated by ▷, showing the first occurence, followed by the text labeled

BAJ:, BBJ:, . . . , of the comment following the "bad" response and the exit to Ⓢ . If a "bad" match has been made a second time, indicated by ∇2, then the acceptable answer is given and the program proceeds to the next question.

Thus the student is allowed only a second chance in each class of "bad" responses. If he makes a second response in the *same class,* he is given the answer. If he goes from a member of one class of "bad" answers to another in a *different class* of "bad" answers, then a new comment is made explaining the reason that new response is unsatisfactory, and he is given a further opportunity to provide an acceptable response. If he makes any two bad responses corresponding to members of the same class then he is given an acceptable answer and the flow passes on to the next question J + 1.

If the student's input is not matched, either by acceptable responses or by unacceptable responses, a comment is presented to him, NAJ:, indicating that his response was not recognized by the program and then the program returns to Ⓢ to accept another input. (NAJ: may contain a hint or an indication of what would be recognized.) If the input is unrecognized a second time, a second hint or suggestion as to what would be recognized is offered him and he is allowed to enter another input. If three responses are made that are not recognized, he is then given the acceptable answer. For unrecognized responses, the current version of this template does not reset the counter GX which records successive good answers. For unrecognized responses the program should give the student the benefit of the doubt and should not conclude that the student is, in fact, unable to offer an acceptable response merely because his input failed to match those that were anticipated.

Figure 2 shows a blank template which could be filled in by a lesson

```
(QJ)_ _ _

          (MGJ)
               (TAJ)_ _ _

          (MBAJ)
               (BAJ)_ _ _

          (MBBJ)
               (BBJ)_ _ _

          (MBCJ)
               (BCJ)_ _ _

          (NAJ)_ _ _ _

          (NBJ)_ _ _ _
```

FIGURE 2 Blank template.

designer and would be processed in accordance with the previous flow diagram for the Jth question. (QJ) indicates the text of the question or statement to be presented by the computer, which invites a response by the student. MGJ labels the class of student inputs that would be considered acceptable "good" responses. TAJ labels the text of an acceptable response which is to be presented before proceeding with the next question. Similarly the other labels correspond to the labels on the previous flow diagram.

Figure 3 shows how such a template was filled in for the first question in a series of fourteen questions on wave propagation, currently operating on computers at the University of Michigan and at the University of California, Irvine. Note here that the first question is a descriptive paragraph with a short question at the end. The "good" match consists of five words, all synonyms for the string referred to in the descriptive paragraph. There is only one class of anticipated *unsatisfactory* responses. It is labeled MBI, and the corresponding output comment is labeled BA1. Note also that one need not have a properly spelled word or even a complete word in order to match with a student's possible input. Consider the word "bump" in MB1, for example. If the b is deleted, then the remaining letters, "ump," would match "hump," "bump," or "lump" (which might be used by the student to describe the shape of the pulse propagating down the string) if any of them appears in the

```
Waves that we see and hear are phenomena in which a *medium*
is subjected by a  source  to a *disturbance* that *propagates*
as time goes on.  Imagine first a taut string which has been
displaced near one end from its straight-line equilibrium position
by a finger (while the ends are held fixed) and released--like
a plucked banjo string.

(Q1)  What is the *medium* in this case?

      (MG1)  'string, rope, cord, thread, wire'
             (TA1)  OK....Here the *medium* in which the disturbance
                    propagates is the string.

      (MB1)  'finger, displacement, bump, push, pull, force, pluck'
             (BA1)  That may give *rise* to the disturbance, but
                    what *undergoes* the disturbance?  Please try
                    again...

      (NA1)  Sorry,...program doesn't recognize your response.
             It recognizes words for the thing that is disturbed.
             Please try again...

      (NB1)  Again doesn't recognize response.  In this case,
             the medium is subjected to a displacement.  What
             undergoes displacement?  One last try...
```

FIGURE 3 Example of a filled template.

student's input. NA1 and NB1 are the output comments to the student, if his input is not recognized.

Figure 4 is the filled-in template for the second question in the program.

```
(Q2)  What is the *source* of the wave?

      (MG2)  'fing, plu, pull, force, push'
             (TA2)  Here the *source* of the wave is the finger that
                    displaces and releases the string.

      (MB2)  'displa, disequil, nonequil, non-equil, distort, ump, pulse'
             (BA2)  That is the *effect* produced by the *source* but
                    what *causes* that effect? Please try again...

      (NA2)  Sorry,...program doesn't recognize your response. It
             recognizes words for the object or agent that causes the
             *disturbance* in the medium. Please try again...

      (NB2)  Again, program doesn't recognize response. In this example,
             the string is distorted by "something." What is it? One
             last try...
```

FIGURE 4 Filled-in template, Question 2.

```
(Q3)  What is the *disturbance* that propagates?

      (MG3)  'displa, pulse, ump, form, shape, distort, force,
             pattern'
             (TA3)  The *disturbance* that propagates here is the
                    distortion of "pulse" produced by the initial
                    displacement of the string.

      (MBA3) 'mass, matter, molecules, string, rope particles, atoms'
             (BA3)  No matter or material propagates from one part of
                    the string to another. Please try again...

      (MBB3) 'energy'
             (BB3)  That *accompanies* the disturbance, but is not
                    the disturbance itself. What *visible* charac-
                    teristic starts at one end and propagates to the
                    other? Please try again...

      (NA3)  Sorry,...program doesn't recognize your response. It
             recognizes words for the characteristic of the medium
             (string) that starts at one end and propagates to the
             other. Please try again...

      (NB3)  Again, program doesn't recognize response. The *distur-
             bance* that propagates is usually a local departure from
             equilibrium. What is that departure for the string
             system? One last try...
```

FIGURE 5 Inadequate responses, Question 3.

Note that it is a simple, short inquiry. The candidates for a "good" match here are the same as were the candidates for a "bad" match in the previous question. The use of the template in Question 2 is the same as that for Question 1. Question 3, however, uses the template slightly differently. This is shown in Figure 5. Here the template has *two* classes of inadequate response. The second class, MBB3, has only one word in it, but if reference is made to "energy," a different comment is made to the student from the comment made if he indicates that "mass" or "matter," etc. propagates down the string.

The template for Question 11, a five-part multiple-choice question, is filled as shown in Figure 6 (we omit the parentheses and numerals). The "good" match consists of 1 and 2 and 5 and "not 4" and "not 3" present. If it fails, then the first class of "bad" matches checks again to see whether the incorrect numbers were omitted from the input. Since the input failed the "good" match, if the "bad" match which follows is satisfied, that would imply that the number of at least one *correct* statement was omitted. If

```
Q:   Imagine that an infinite plane is passed through the
     line along which the wave propagates.  If such a plane
     can be found that also contains the *displacement
     pattern as it propagates*, then the wave is said to be
     *polarized* in the direction of the displacement.  Type
     in the *numbers* that label the statements below which
     you believe to be *correct*.

     (1)  Sound waves are longitudinally polarized.
     (2)  The pulse on the string described above is polar-
          ized transverse to the direction of propagation.
     (3)  If one end of a long wire (which is fixed at the
          other end) is quickly twisted and released, a
          polarized pulse is propagated along the wire.
     (4)  The wind-driven waves on a wheat field (the "amber
          waves of grain") are polarized in some direction.
     (5)  The expanding circle of ripples that are propa-
          gated when a pebble is thrown into a placid pond,
          are polarized waves.

MG:   1 & 2 & 5 & N3 & N4

MBA:  N3 & N4

MBB:  (3,4) & 1 & 2 & 5

MBC:  (3,4)

TA:   1, 2, and 5 are considered correct statements...

BA:   At least one correct statement omitted.  Please try
      again.
BB:   At least one incorrect statement included.  Please
      try again.
BC:   At least one correct statement omitted and at least
      one incorrect statement included.  Please try again.
NA:   Sorry.  Program recognizes numbers only here.  Please
      try again.
NB:   Again, doesn't recognize response.  Please use some
      combination of 1, 2, 3, 4, 5.  One more try...
```

FIGURE 6 Multiple choices, Question 11.

neither of the matches MG or MBA is satisfied, then MBB checks to see if one or the other of the incorrect statement labels appeared together with all the correct ones. If so, that is a different kind of unsatisfactory input from those contained in MBA. Lastly, MBC is the match corresponding to at least one of the incorrect statement labels present and also at least one of the correct statement labels missing. The alternative text outputs for the various responses are listed below them.

A more ambitious use of templates, with possible implications for dependent branching within each question, is being developed in which several inputs may be requested from the student in a particular order. Thus, for a question in the form

Q: Consider now a wave that we do not detect with our senses. A radar pulse sent to outer space. . . for such a wave type on one line the words that fill in the blanks below in the same order as A, B, C, D separating each answer with a comma and ending with a period.

A) The source of the disturbance is ________.
B) The disturbance is ________.
C) The direction of polarization is ________.
D) The energy transported is in __ form ________.

A good match would consist of one string from each of the four groups in parenthesis in the statement:

MG: (charge, electron, current, anten, transmi) & (electr, magn, field, feild) & (transv, rt & ang, perp) & (electr, magn, field, feild)

Words or parts of words indicative of an incorrect response are given by

MBA: longit, para, mech, kine, poten,

and appropriate texts are added. This approach is used for Question 14, the last of the wave series. A typical student conversation in which all fourteen questions are considered is found in the Appendix.

A separate set of programs exists that prompts the lesson designer. It is known as DITRAN and is the subject of a paper submitted for publication elsewhere. The set of programs is written in a language called CRLT-PIL, which is an extension of PIL. This version of DITRAN is available on the IBM 360/67 at the University of Michigan. The system is not exportable without a great deal of individual work, since CRLT-PIL is not widely known. Further inquiries are welcome and may be directed to me or to the Center for Research in Learning and Teaching at the University of Michigan.

At the present time, we are developing more elaborate templated structures at the Lawrence Hall of Science. These advanced templates are being used in the design of physics dialogues intended for high school physics teachers. It is still too early to evaluate the effectiveness of the more complex dialogues that we have produced. It is safe to say, however, that the early users find them absorbing, and they provide a source of indications for further improvement. We hope that in about a year we shall be able to make more definitive statements about templated dialogue in physics.

The opportunity to design dialogues composed of these "templated" exchanges was offered to participants in the Tuesday evening workshop at the conference. The starting group of about 20 people elected to write such a dialogue on Gauss's Law. A total of 48 questions, applying the template in various ways were written, and 36 of these were usable as working dialogues before the end of the conference. (As expected, in consequence of the meagre time available for lesson design, the pedagogy left much to be desired but the computer and the templated programs did *their* parts admirably.) On the whole templates appear quite feasible as a device for authors to circumvent the problems of programming in the development of computer-based dialogues. Further details on the activities of the workshop are available from the Commission on College Physics in the form of a report entitled "Templates for the Construction of Computer-Based Self-Instructional Dialogs: Gauss's Law," by C. D. Bennett, B. F. Gore, G. T. Johnston, and N. Sherman.

APPENDIX: A TYPICAL CONVERSATION

WAVES THAT WE SEE AND HEAR ARE PHENOMENA IN WHICH A *MEDIUM* IS SUBJECTED BY A *SOURCE* TO A *DISTURBANCE* THAT *PROPAGATES* AS TIME GOES ON. IMAGINE A TAUT STRING WHICH HAS BEEN DISPLACED NEAR ONE END FROM ITS EQUILIBRIUM POSITION BY A FINGER (WHILE THE ENDS ARE HELD FIXED) AND RELEASED–LIKE A PLUCKED BANJO STRING.

Q(1) WHAT IS THE *MEDIUM* IN THIS CASE?

?TAUT CORD
OK
HERE THE *MEDIUM* IN WHICH THE DISTURBANCE PROPAGATES IS THE STRING.

Q(2) WHAT IS THE *SOURCE* OF THE WAVE?

?FINGER PLUCK
GOOD...
HERE THE *SOURCE* OF THE WAVE IS THE FINGER THAT DISPLACES AND RELEASES THE STRING.

Q(3) WHAT IS THE *DISTURBANCE* THAT PROPAGATES?

?VIBRATIONS
VERY GOOD...
THE *DISTURBANCE* THAT PROPAGATES HERE IS THE DISTORTION OR *PULSE* PRODUCED BY THE INITIAL DISPLACEMENT OF THE STRING.

Q(4) SUPPLY THE MISSING WORD IN THIS SENTENCE: THE DIRECTION OF THE *DISPLACEMENT* OF THE STRING IS ______ TO THE EQUILIBRIUM LINE OF THE STRING.

?ALONG
ASSUME THAT THE STRING IS COMPLETELY INELASTIC SO THAT NO DISPLACEMENT PARALLEL TO THE STRING IS POSSIBLE. PLEASE TRY AGAIN...

?ORTHOGONAL
SORRY... PROGRAM DOESN'T RECOGNIZE YOUR RESPONSE. IT RECOGNIZES WORDS FOR THE DIRECTION OF THE INITIAL DISPLACEMENT RELATIVE TO THE DIRECTION OF THE UNDISTURBED STRING. PLEASE TRY AGAIN...

?RT ANGLES
AGAIN DOESN'T RECOGNIZE RESPONSE. THINK OF THE ANGLE MADE BY THE FORCE THAT PRODUCES THE INITIAL DISTORTION OF THE STRING AND THE STRING ITSELF. ONE LAST TRY...

?90 DGRE#
90 DEGREES
OK
THE DISPLACEMENT IS PERPENDICULAR TO THE STRING.

Q(5) THE DIRECTION OF *PROPAGATION* OF THE DISPLACEMENT PATTERN IS ______ TO THE EQUILIBRIUM LINE OF THE STRING.

?ALONG
GOOD...
THE DISPLACEMENT PROPAGATES IN THE DIRECTION PARALLEL TO THE STRING.

Q(6) NOW IMAGINE A PULSE OF SOUND WHOSE SOURCE WAS AN EXPLODED FIRE-CRACKER. WHAT IS THE *MEDIUM* THROUGH WHICH THE SOUND PULSE PROPAGATES?

?ATMOSPHERE
VERY GOOD...
THE *MEDIUM* HERE IS AIR. SOUND PROPAGATES THROUGH GASES, LIQUIDS, AND SOLIDS.

Q(7) ANY OF SEVERAL WORDS CAN BE USED TO DESCRIBE THE *DISTURBANCE* THAT PROPAGATES IN A SOUND WAVE; TRY A FEW...

?NOISE
SORRY... PROGRAM DOESN'T RECOGNIZE YOUR RESPONSE. IT RECOGNIZES WORDS FOR THE *DISTURBANCE* THAT PROPAGATES THROUGH THE MEDIUM. PLEASE TRY AGAIN...

?COMPRESSED AIR
FOUR STRAIGHT, SO FAR...

THE *DISTURBANCE* THAT PROPAGATES IN SOUND WAVES IS LOCAL *FLUCTUATIONS IN DENSITY AND PRESSURE*. COMPRESSIONS AND RAREFACTIONS ARE OTHER WORDS DESCRIBING THE FLUCTUATIONS.

Q(8) THE *DIRECTION* OF THE DISTURBANCE (I.E. DISPLACEMENTS) IS ______ TO THE DIRECTION OF PROPAGATION OF SOUND WAVES.

?RIGHT ANGLES
TRANSVERSE DISTURBANCES PROPAGATE THROUGH SOLIDS, BUT THINK ABOUT THE MECHANISM OF PROPAGATION OF SUCH A DISTURBANCE THROUGH A LIQUID OR A GAS. PLEASE TRY AGAIN...

?PARALLEL
OK
SOUND WAVES ARE COMPRESSIONS AND RAREFACTIONS THAT DISPLACE THE MEDIUM IN DIRECTIONS *PARALLEL* TO DIRECTIONS OF PROPAGATION.

Q(9) WHAT IS THE WORD USED TO CHARACTERIZE WAVES THAT PROPAGATE IN DIRECTIONS *PERPENDICULAR* TO THE DIRECTION OF THE DISTURBANCE?

?ORTHOGONAL
SORRY. . . PROGRAM DOESN'T RECOGNIZE YOUR RESPONSE. IT RECOGNIZES WORDS FOR THE SPECIAL KIND OF WAVE WHOSE DISTORTION PATTERN IS PERPENDICULAR TO THE DIRECTION IN WHICH THE PATTERN MOVES. PLEASE TRY AGAIN...

?TRANSVERSE
GOOD...
SUCH WAVES ARE CALLED *TRANSVERSE* WAVES.

Q(10) WHAT IS THE WORD FOR A WAVE PROPAGATING *PARALLEL* TO THE DISTURBANCE?

?LONGITUDINAL
VERY GOOD...
SUCH WAVES ARE CALLED *LONGITUDINAL* WAVES.

Q(11) IMAGINE THAT AN INFINITE PLANE IS PASSED THROUGH THE LINE ALONG WHICH THE WAVE PROPAGATES. IF A PLANE CAN BE FOUND THAT ALSO CONTAINS THE *DISPLACEMENT PATTERN AS IT PROPAGATES*, THEN THE WAVE IS SAID TO BE *POLARIZED* IN THE DIRECTION OF THE DISPLACEMENT. TYPE IN THE *NUMBERS* THAT LABEL THE STATEMENTS BELOW WHICH YOU BELIEVE TO BE *CORRECT*.

... (1) SOUND WAVES ARE LONGITUDINALLY POLARIZED.
... (2) THE PULSE ON THE STRING DESCRIBED ABOVE IS POLARIZED TRANSVERSE TO THE DIRECTION OF PROPAGATION.

... (3) IF ONE END OF A LONG WIRE (WHICH IS FIXED AT THE OTHER END) IS QUICKLY TWISTED AND RELEASED, A POLARIZED PULSE IS PROPAGATED ALONG THE WIRE.
... (4) THE WIND-DRIVEN WAVES ON A WHEAT FIELD (THE 'AMBER WAVES OF GRAIN') ARE POLARIZED IN SOME DIRECTION.
... (5) THE EXPANDING CIRCLE OF RIPPLES THAT ARE PROPAGATED WHEN A PEBBLE IS THROWN INTO A PLACID POND, ARE POLARIZED WAVES.

?124
AT LEAST ONE CORRECT ANSWER OMITTED AND AT LEAST ONE WRONG ANSWER INCLUDED. PLEASE TRY AGAIN...

?12
AT LEAST ONE CORRECT ANSWER OMITTED. PLEASE TRY AGAIN...

?123
1, 2, AND 5, ARE CONSIDERED CORRECT STATEMENTS.

Q(12) AS A DISTURBANCE PROPAGATES IT 'CARRIES' *ENERGY* WITH IT. IN THE CASE OF A DISPLACEMENT PULSE TRAVELING ALONG A STRING AGAIN, THE ENERGY ASSOCIATED WITH THE PROPAGATING PULSE IS IN THE FORM OF______ _____. (TWO WORDS, PLEASE...)

?KINETIC EN
PART OF THE ENERGY APPEARS IN YOUR ANSWER; ALL OF IT CAN BE DESCRIBED BY TWO WORDS (ONE OF WHICH IS 'ENERGY'...). PLEASE TRY AGAIN...

?POTEN EN
MECHANICAL ENERGY IS TRANSFERRED WITH THE PULSE; THE POTENTIAL ENERGY OF STRETCHING AND THE KINETIC ENERGY OF THE PARTS IN MOTION.

Q(13) THE PREVIOUS EXCHANGE DESCRIBED THE SENSE IN WHICH ENERGY IS TRANSFERRED BY A WAVE FROM ONE PART OF A MEDIUM TO ANOTHER. NOW CONSIDER WHETHER MATTER (A COLLECTION OF MOLECULES) IS TRANSFERRED FROM ONE PART OF A MEDIUM TO ANOTHER WHEN A DISTURBANCE PROPAGATES THROUGH IT.

TYPE IN THE NUMBERS OF THE STATEMENT BELOW WHICH YOU CONSIDER TO BE CORRECT.

... (1) IN A TRANSVERSE PROPAGATING WAVE, MATTER IS NOT TRANSFERRED ALTHOUGH ENERGY IS.
... (2) IN A LONGITUDINAL WAVE, PARTICLES MAY BE MOVED IN THE DIRECTION OF PROPAGATION, BUT WHEN THE MEDIUM HAS RETURNED TO ITS UNDISTURBED CONDITION, ENERGY HAS BEEN TRANSFERRED THROUGH THE MEDIUM WHILE MATTER HAS NOT.
... (3) STATEMENT (2) IS ALSO TRUE FOR A WAVE THAT HAS A DISPLACEMENT WITH BOTH TRANSVERSE AND LONGITUDINAL COMPONENTS.

. . . (4) A SURFBOARD MAY MOVE WITH A WATER WAVE, BUT IN THE REGION WHERE THE WAVE HAS NOT BROKEN, NO APPRECIABLE WATER HAS BEEN TRANSFERRED WITH THE SURFBOARD.

?124
AT LEAST ONE CORRECT ANSWER OMITTED. PLEASE TRY AGAIN. . .

?1234
OK
ALL FOUR STATEMENTS ARE CONSIDERED CORRECT.

Q(14) CONSIDER NOW A WAVE THAT WE DO NOT DETECT WITH OUR SENSES. . . A RADAR PULSE SENT TO OUTER SPACE. . . FOR SUCH A WAVE, TYPE ON ONE LINE THE WORDS THAT FILL IN THE BLANKS BELOW IN THE SAME ORDER AS A, B, C, D, SEPARATING EACH ANSWER WITH A COMMA AND ENDING WITH A PERIOD.
. . . (A) THE SOURCE OF THE DISTURBANCE IS ______.
. . . (B) THE DISTURBANCE IS ______.
. . . (C) THE DIRECTION OF POLARIZATION IS ______.
. . . (D) THE ENERGY TRANSPORTED IS IN ______ FORM.

?ANTENNA, ELECTRIC FIELD, ARBITRARY, ELECTROMAGNETIC.
GOOD. . .
A='ACCELERATED CHARGES', B='ELECTROMAGNETIC', C='TRANSVERSE', D='ELECTROMAGNETIC'.

>>>>>>

THAT'S ALL THIS PROGRAM HAS READY RIGHT NOW. . .
BEFORE YOU LEAVE, PLEASE TYPE IN YOUR OPINIONS ABOUT IT.
DID IT HELP YOU LEARN ANY PHYSICS?
DID IT RUN SMOOTHLY?
WHAT SIMPLE IMPROVEMENTS IN PROGRAMMING WOULD MAKE LEARNING MORE EFFECTIVE?
(TO TYPE SEVERAL LINES, BE SURE TO END EACH LINE AS USUAL AND TYPE 'ZZZ' BEFORE ENDING LAST LINE SO THAT PROGRAM WILL NOT EXPECT ANOTHER INPUT LINE.)

?IT DOESN'T SEEM TO KNOW WHEN I'M WRONG OR JUST JOKING
?LOOK AT 14. ALSO SHOULD RECOGNIZE NOISE AND ORTHOGONAL ZZZ
THANKS FOR YOUR REMARKS. . . COME BACK AGAIN. . .
#EXECUTION TERMINATED

CAI in a Multimedia Physics Course

ANTON F. VIERLING and
ARISTIDES T. SERLEMITSOS

INTRODUCTION

The changing scene in American education results from the influence of a number of opposing forces: the enormous increase in knowledge and skills that must be acquired by students is limited by the time frame of our typically lockstep curriculum; a growing population of eager students is often neglected due to the dwindling supply of competent teachers; technology intrudes with new devices for education, but school administrators must contend with minimum budgets to support existing facilities; an ever-changing employment market requiring new skills confronts the conservatism of educators and their reluctance to change. The organization, the curriculum, the techniques and even the schools are slowly evolving through these socioeconomic forces that are both the cause and result of change.

There may be substantial reasons for the academic community's reluctance to change teaching and administrative techniques. To begin with, what guidelines does one have for making instructional decisions and/or changes? For instance, what medium (or media mixes) are most effective in teaching of specific kinds of course objectives? What specific student learning characteristics require additional attention and correction? These and similar questions led to a research project conducted at the U.S. Naval Academy in conjunction with the U.S. Office of Education.*

The scope of the research was to evaluate the teaching effectiveness of various educational media in three fields of undergraduate study; namely, physics, economics, and psychology. The research on each topic was planned for three years' duration and can be roughly divided into three phases: preparation, validation and evaluation. The authors of this manuscript were part of a team of educators who together with the New York Institute of

*The Project Manager of the U.S. Naval Academy is Dr. Jesse Koontz, who is also the director of the Educational and Management System Center. Dr. Richard Otte is the Project Officer for the Office of Education.

Honeywell Educational Resource Center, Minneapolis, Minnesota. Quality Education Development Corporation, Boston, Massachusetts.

Technology (subcontractor) and the U.S. Naval Academy worked on the experimental physics course. The physics course content included basic undergraduate mechanics, electricity, and magnetism.

The purpose of this manuscript is to report on one portion of the total research project, specifically, that which involved the computer as a means of communicating instruction to students through problem-drill and reinforcement. The manuscript briefly describes the experimental design and reports on the progress through January 1970. For purposes of discussion, some quantitative student data are presented. The detailed statistical comparison of media effectiveness and final reporting is still in progress. Although this educational research project is not yet complete, some interim reports and documentation which have been printed are listed at the end of this paper.

DESCRIPTION OF EXPERIMENTAL MEDIA

In the context of education, the word medium refers to the intermediate that acts between a learning objective and the student who is striving to achieve that learning objective. The learning medium may take the form of the spoken word, the written word, or any of a number of devices that stimulate the learning process. If one examines the existing learning media, it becomes obvious that the same input to the senses can be stimulated by various media with markedly different responses. For example, if the learning objective were to describe wave motion we could expect that a photograph, a silent movie, and a computer-animated film of an ocean wave would evoke varying responses from the student. That is, the sensory input may be activated by different levels of stimuli, depending upon the medium used to achieve a particular learning objective.

Can one actually measure the effectiveness of a stimulus or a mix of stimuli in achieving specific learning objectives? The approach to this question which we adopted for the current study was to select a specific subset of physics course objectives which would be communicated to students by various media in order to measure the relative effectiveness of each. The criteria for selection of the specific course objectives to be used in the experimental media were as follows:

(1) The concept or principle illustrated was judged to be dependent on visulized motion for effective learning.
(2) The concept or principle illustrated was of fundamental importance in the course.
(3) The concept or principle was determined from teaching experience to be difficult for students to learn.
(4) The materials were selected from different content topics in the course.

Obviously, a considerable amount of personal bias is involved in such a selection process; however, in all cases a consensus of the subject matter experts was obtained.

Having established a subset of twenty measurable learning objectives from the course (Table 1), media were chosen partly on the basis of ability to isolate certain stimulus-response characteristics. Specifically, an attempt was made to choose the stimulus to be studied—i.e., audio, visual, or sensory—and then to select a medium which presented the desired stimulus or stimuli. The selection of a specific medium was dictated partly on anticipated developmental effort and cost. It was assumed, for instance, that audio-visual television tapes present the same stimuli as motion picture film and could be developed at lower costs.

In addition to the six basically different media selected for investigation—programmed instruction, video tape, audio tape, photographs, computer-assisted instruction, and lecture-demonstration—four textbooks common to all students were chosen for the course, thus locking in a significant amount of information and content to be provided by the printed word. This, of course, precluded the possibility of providing information dealing with certain objectives by audio only or by audio with pictures, with no access to the printed word. It was also decided that all students would witness the same physics laboratory, rather than for one group to do special laboratory routines or no laboratory at all.

A. The Study Guide Medium (SG)

Since textbooks were available to all students, a principal consideration was the student's method of interacting with the textbooks. The Study Guide was designed to direct the interaction and cause the student to respond actively with the instructional information. The Study Guide is a programmed textbook which contains over 1100 questions in sixteen separately bound volumes, one for each week of the semester. The questions and answers are scrambled in such a manner that the student can only progress through the volume by committed responses on a latent-image matrix, i.e., he must mark an answer page with a special pen to discover what text page is next. The student is presented with multiple-choice questions, each choice directing him either to a different page in the scrambled volume for extensive remedial text, or, if his answer is correct, to the next question.

The Study Guide depends on the four physics textbooks for reading assignments and has been planned to communicate all of the physics course objectives including core, review and enrichment. A hierarchy of learning has been established for each Study Guide component such that the learning categories can be identified as (1) recall, (2) mathematical manipulation, (3) synthesis of solution, and (4) multiple step. The stimulus-response format of

TABLE 1 Subset of Learning Objectives for Mixed-Media Presentation

Course Objective Number	Description
14	Apply Newton's second law of motion.
23	Answer fundamental questions relating to the physical significance of the kinetic energy of the body.
25	Answer fundamental questions and solve problems relating to the nature of mechanical potential energy and the units used to measure it.
27	Use the concept of conservation of mechanical energy to solve various problems in kinematics where conservation of mechanical energy is applicable.
30	Solve momentum problems involving bodies with constant mass.
32	Solve momentum problems involving bodies with variable mass.
36	Answer verbal questions and analyze situations dealing with two-body collisions.
41	Solve problems based on Newton's law of universal gravitation.
46	Answer fundamental questions and solve problems based on the concept of the gravitational field strength.
53	Calculate the electric field for any charge distribution.
61	Calculate the trajectory of a charged particle traveling in an electric field.
67	Apply Gauss's law.
78	Answer questions and solve problems involving the capacitance of a parallel-plate capacitor having two plates each of Area A, separate by a distance d in vacuum.
80	Solve descriptive and numerical problems involving capacitors in series and parallel combinations. (Note: all interconnecting wires are resistanceless.)
92	Answer questions relative to the methods of application of Kirchhoff's rules to electric networks.
93	Apply Kirchhoff's rules to the solution of numerical problems ranging from simple to more complex multiloop networks.
108	Use the mathematical statement of Ampere's law in the solution of qualitative and quantitative problems.
110	Analyze problems involving the force between two parallel wires as given by $F = M_0 L i_b i_a / 2\pi d$.
112	Use the expression for the magnetic induction within an ideal solenoid to solve quantitative and qualitative problems.
115	Apply Lenz's law to determine the direction of an induced emf in various induction situations.

the Study Guide is the printed word in a highly structured and organized form. The mode of communication is silent reading with active responses made by the student and immediate feedback as to successful progress. The branched structure of the Study Guide allows the student to by-pass certain portions depending upon his comprehension of the written text.

B. The Audio-Visual Medium

Some topics of physics, kinematics, for example, require the student to describe formally the motion of an object. One might conjecture that a student would learn more readily if he could actually see the object in motion as the concepts are being explained to him. The conventional technique for presenting students with such a visual experience is through a classroom demonstration. When given "live," such a session allows a student to use all of his senses for stimulus-inputs. Questions can be asked and parts of the experiment can be repeated, if necessary. One group of students was exposed to these demonstrations and had access to the study guide (L/SG).

The demonstration can also be filmed for later exposure to students. How much is lost because students cannot question the teacher immediately is not known. The video tape recorder was chosen as a presentation medium to communicate to the student learning objectives that could include a moving visual stimulus. Each audio-visual (AV) tape consists of approximately 20 to 30 minutes of demonstration plus verbal explanation. The associated research hypothesis here would be that when the learning objective requires the student to respond to motion, to visualize motion, or to produce motion, then appropriate motion as part of the relevant instructional visual stimuli should facilitate learning.

The student is thus exposed to a learning environment which includes motion, voice and still pictures.

C. The Talking Book Medium (TB)

To examine the necessity of actually visualizing motion in achieving a learning objective the medium selected must show the phenomenon being studied by a picture or graphical representation (no motion). In order to accomplish this, still pictures were made of the audio-visuals of subsection B and these were organized into a highly illustrated book. Students used an audio tape recorder with voice explanations of all the still pictures in the book. Given the same set of learning objectives, seeing motion is perhaps not essential, but only helpful. The still pictures provide a thorough replay of the event, while a voice explains the physics concepts involved. The voice directs the student's attention to key items in the visuals with the minimum of distraction.

The student's learning environment is now a combination of voice and still pictures but no motion.

D. The Illustrated Book Medium (IB)

The Illustrated Book contains the same illustrations as the Talking Book; however, the spoken words from the audio-tape have been transcribed into printed text. In this case, the student's visual attention is divided between reading the printed information and looking back and forth from the printed page to the picture.

The student is exposed to a learning experience with no voice and no motion but with many still pictures and the printed word.

E. The Computer-Assisted-Instruction Medium (CAI)

The computer, as a means of achieving learning objectives, presents some rather unique alternatives. At least two categories of computer implementation become apparent to subject-matter experts responsible for the design of media strategy. First, the computer can be an easily programmable drill-and-practice device. A learning objective in physics is often tested by asking the student to display a certain skill. This may be simple recall of terminology, a mathematical manipulation, or, perhaps, a multiple synthesis of a solution. The computer was, therefore, programmed to drill the student on these skills. The student, at his convenience, could work in his area of difficulty by accessing the time-shared computer and requesting one of many problem drills, each dealing with a specific type of problem, e.g., free fall, inclined plane, electric field, etc.

The following characteristics were considered important in design of these computer programs:

(a) students could choose the area in which they needed particular assistance;
(b) all answers were immediately followed by an indication of right or wrong;
(c) wrong answers were followed by an explanation of the physical problems involved;
(d) repetition was avoided and programming ingenuity was used to enhance student interest;
(e) most programs were designed to be approximately 30-minute sessions.

The computer has the advantage of randomly changing the problem type and known variables to maintain a continuous drill for the student. Hidden answers were common, and internal branching was programmed to be dependent upon student response. A record of student progress was kept by the computer for subsequent management processing.

Another equally important role of the computer in the learning process appears to be in problem-solving. The merging of computer programming skills with basic physics concepts has encouraged students with very little training in mathematics to solve problems of increasing physical complexity. It was anticipated that the student who "teaches" the computer to, for instance, predict the *X*- and *Y*- coordinates of a projectile's trajectory will better understand and remember the mechanics of a projectile problem. For this portion of the computer involvement, the material was arranged into individual lessons, with each lesson planned to be one week of a student's study time for a particular learning objective. A sample lesson plan follows:

> *TOPIC:* Time of Flight
> Physics Course Objective: 14
> *CAI Lesson Objective:* At the completion of this lesson you should be able to write a computer program that predicts the time of flight of a projectile fired at any angle.

The CAI medium was designed to saturate the stimulus-response environment. The computer terminal became a resource for the student in both problem drill and problem solving.

F. Controls—Lecture/Demonstration (L/D)

A conventional undergraduate physics class has at least one or more of the following characteristics: (1) a set of course objectives; (2) a teacher to explain concepts, interpret the textbook, and assist in problem solving; and (3) an array of demonstration devices which may be used at the discretion of the teacher. This might best describe the learning experience of the control students (CNTR). The teacher knew what had to be taught and used everything at his disposal to reach that objective restricted only by time, energy and incentive. The teacher could not, however, use any of the material specifically prepared for the other media groups.

H. Student Option (SO)

Another and final group was referred to as Student Option (SO). These students were also a control group in the sense that they were completely free to sample from one or all of the media available. They could attend a lecture, work on the Study Guide, view a video tape, and so on. All use of media was recorded and is being analyzed to establish a possible student preference or to isolate individual learning styles.

STUDY DESIGN AND INTERIM RESULTS

In order to obtain some statistical measure of the effectiveness of the media described in the previous section, the students were randomly divided into eight groups of 26 students. Each group was assigned to a different one of the media conditions each week, including a lecture-Study Guide mix (L/SG). The CAI medium was not rotated between groups since it was felt that learning to use the computer terminal plus programming techniques could not be accomplished and implemented in a week. Therefore, the group of students assigned to CAI did *not* rotate between media conditions.

TABLE 2 Weekly Assignment of Groups to Media

MEDIA	WEEK															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AV	A	E	B	D	C	G	F	Review 3 Week	A	F	C	E	G	B	D	Final Exam
TB	B	C	G	E	A	F	D		C	A	E	G	B	D	F	
IB	C	D	A	F	E	B	G		G	E	B	D	F	A	C	
L/SG	D	G	E	B	F	C	A		E	C	G	B	D	F	A	
SG	E	F	C	G	D	A	B		F	D	A	C	E	G	B	
L/D	F	B	D	A	G	E	C		D	B	F	A	C	E	G	
SO	G	A	F	C	B	D	E		B	G	D	F	A	C	E	
CAI	H	H	H	H	H	H	H		H	H	H	H	H	H	H	

The assignment of student groups to media conditions each week was accomplished by generating a Latin Square for the first seven weeks and a Latin Square for the second seven weeks, assigning groups to blocks in a 7 by 7 matrix (Table 2) with the requirement that there be a different group in each condition each week, and no group be assigned to any condition more than once. A table of random numbers was used to determine the order of the rows and columns. The purpose of randomizing the sequence in which each group receives the various conditions is to cancel out possible effects due to the *order* of experience with different media effects. In Table 2, the sequence that each group followed through various conditions can be determined by tracing the same letter across the columns. Each letter identifies a specific student group, and the designation remained the same throughout the sixteen-week course.

On the same day each week, all groups received a pre-test and a post-test to measure the achievement of course objectives covered that week. The pre-test made it possible for some students to by-pass the following week's work. The post-test was designed to indicate specifically which students were in trouble so that remedial action could be applied. The computer was used extensively in student record keeping, locating students with recurring problems, and assigning remedial work based on performance.

Due to the intricacies of an educational experiment, many variables must remain almost uncontrolled. The fact that students take other courses together, exchange information for study purposes, and generally are encouraged to acquire knowledge and skills through all possible means, makes evaluation studies difficult. In addition, when dealing with real "live" students, one cannot, in good conscience, design an experimental medium which might in any way jeopardize a student's ability to achieve a certain course objective. Students naturally object to being part of a treatment group when their study-time is at stake.

The effect of these reflections upon the experimental course design and implementation was to place unusual emphasis upon the content of each

TABLE 3 Percentage of Students Failing to Achieve Course Objective

Course Objective Number	Breakdown of Failures by Percentage of Each Media Group							
	CAI	SG	AV	TB	IB	L/SG	SO	CNTR
14	0	0	5	0	18	6	6	0
61	0	24	31	46	30	33	54	15
67	0	6	5	6	5	4	15	5
70	3	0	0	4	0	0	0	0
78	17	21	15	19	21	35	19	24
25	22	22	19	11	0	9	0	4
92	27	55	43	36	50	48	59	45
110	31	18	30	23	20	21	7	45
112	32	41	45	64	47	42	53	55
93	33	22	29	22	25	35	14	20
53	33	25	35	11	15	21	25	24
30	41	15	8	14	5	5	0	11
23	43	61	67	59	40	43	53	54
27	43	39	48	44	40	22	26	21
32	58	30	19	27	27	46	26	28
36	58	30	0	32	9	14	16	50
41	75	33	38	39	50	38	36	48
46	75	71	48	50	64	62	27	61
115	75	82	75	82	67	89	73	77
108	85	47	80	68	67	63	67	68

teaching medium. The student had to be given the enabling tools to achieve the final objective. By the weekly pre-test and post-test design, students with difficulties could be isolated for an in-depth tutorial to prepare them for the subsequent week of new objectives. Furthermore, the twenty course objectives chosen for the experimental media represented only one sixth of the total number of objectives treated during the sixteen weeks.

The results of student performance on specific course objectives have been arranged in order of decreasing performance by the CAI group. Upon inspection of the percentages in Table 3, one sees that, with few exceptions, the general trend of decreasing performances appears to be parallel in the other media. This would indicate some uniformity in the degree of difficulty of objectives as presented by the various media. The exceptions may prove to be more interesting than the rule in this study. For instance, course objective number 61 was unanimously achieved by CAI students but consistently missed by the others. The CAI students were asked to program the general projectile problem in mechanics (objective 14), and perhaps this reinforcement assured retention of the concept until the discussion of electric forces and fields. On course objective 25, a reverse situation seems to be indicated. The CAI students received drill-and-practice problems, but they were outperformed by all the other groups.

In order to obtain an overview of these experimental results, the course objectives were grouped in three categories. This grouping related directly to the initial choice of these objectives in the experiment; namely,

(a) achievement of the objective was judged to be dependent upon visualizing motion;

(b) the objective was deemed of fundamental importance to the course;

TABLE 4 Percentage of Students Failing to Achieve Course Objectives by Categories: (A) Dependent upon Visualizing Motion; (B) Fundamental to the Course; (C) Difficult Objectives

MEDIA	CATEGORIES		
	A	B	C
CAI	37	29	46
SG	34	24	52
AV	32	30	43
TB	38	24	45
IB	28	25	57
L/D	31	28	50
SO	31	26	32
CNTR	32	29	55

(c) achievement of the objective was judged difficult from prior teaching experience.

In Table 4 one can observe the performance of students in each group on the three categories of objectives. The numbers now represent percentages of students who failed to achieve the category of objective. It would appear that category A (motion dependence) was equally successful in all media. Category B (fundamental importance) was also well represented by each medium. Category C (historically difficult) proved to be difficult again, but perhaps, the Student Option group has the advantage.

These data are presented primarily for discussion purposes, not as final conclusions. A great deal of statistical work is now in progress to study the many interrelating factors of the experiment and to determine the possible existence of subtle differences in student performance. The computer is being used extensively in these final correlations and tabulations of data.

REFERENCES

1. W. A. Deterline and R. K. Branson, *Evaluation and Validation Design*, Technical Report #4.7, U.S.O.E. Project #8-0446, November 3, 1969.
2. W. A. Deterline and R. K. Branson, *Design for Selection of Strategies and Media*, Technical Report #4.9, U.S.O.E. Project #8-0446, October 20, 1969.
3. A. F. Vierling and A. T. Serlemitsos, *Course Development and Restructure*, Technical Report #4.20, U.S.O.E. Project #8-0446, November 3, 1969.
4. A. F. Vierling and A. T. Serlemitsos, *Implementation Design*, Technical Report #4.14, U.S.O.E. Project #8-0446, November 3, 1969.
5. A. F. Vierling, *CAI Development of Multimedia Physics*, Technical Report #4.30, U.S.O.E. Project #8-0446, November 3, 1969.

Instructional Software Engineering

C. VICTOR BUNDERSON

The burden of this paper is that an emerging discipline of *instructional software engineering*, sometimes referred to as "instructional design," is a critical component in the full development of computer uses in education. This discipline will be a synthesis and extension of parts of several existing fields: *Instructional* implies education disciplines, especially educational psychology, involving human learning, psychometrics, programmed instruction, and individual differences. *Software* implies computer science subdisciplines, especially system analysis, programming, artificial intelligence, natural language processing, and computer graphics. *Engineering* implies a problem-oriented design science approach—an iterative, empirically oriented pragmatic approach. Instructional Software Engineering, then, is the technology of instructional materials, their design and development, and the future instructional theory behind this technology. As much as the hardware, it has the potential of changing education from the labor-intensive, low yield field it now is to a man-machine system possessing probably a lower teacher-pupil ratio but with a far higher yield in terms of educational accomplishments and values on the part of the students.

ADJUNCT AND MAINLINE USES

Such a transformation cannot come so long as computers remain on the periphery of education. The adjunct uses of computers by teachers as supplements, laboratories, homework, etc., which dominate this conference, are having an important impact on education but should be viewed in contrast to computer applications which take on a mainline instructional burden. By adjunct is meant those applications of computers that are supplementary, such as problem solving, simulation and modeling, demonstrations, drill, and review. A mainline CAI application, however, is a complete system to teach,

Director, Computer-Assisted Instruction Laboratory, University of Texas, Austin, Texas.

at the least, a complete course. Some of the distinctions between these two classes of application are listed below:

ADJUNCT →	MAINLINE
Context provided by teacher	Complete man-machine system
Programming by teacher and student or informal exchange	Specifications and programming by design-production teams
Represents an add-on cost	Great economic potential—supplantive
Requires low to moderate capital investment	Requires high capital investment

Other Dimensions for Comparison

Increased effectiveness: opportunity for restructuring objectives and subject matter	Increased effectiveness and efficiency
Modest but variable system requirements	Specific engineering design
TTYs or typewriters	CRT or Plasma Display
Batch or interactive → sophisticated graphics	Interactive and efficient
Standard languages	Special author & Student languages
Fits within standard credit-hour scheduling	Requires self-paced scheduling and grading

The fact that adjunct program development can lead to "mainline" use of computers is recognized by the arrows leading from the adjunct end of the continuum to the mainline end. If enough supplementary materials are produced over a period of time, they could be organized into a teaching system.

Considering CAI as a new technology rather than as another kind of medium to be used by the teacher in his traditional classroom has important consequences in terms of staffing, costs and effectiveness, and the organization and management of educational systems. Mainline applications will be more cost-effective than adjunct but will require higher capital investment in hardware and instructional design and development. Also, mainline use will require a self-paced, individualized scheduling system and an objective-oriented conception of grading.[1]

On the other hand, adjunct applications will often be more innovative than a total system engineered and produced for wide distribution. The individual faculty member and his students, especially in higher education, can use the computer in modeling, rethinking, and restructuring a portion of their field. This restructuring can have a major influence on the design of the next generation mainline program in that subject.

Good teachers immersed in teaching and research not only have this potential for restructuring in the adjunct mode, but regard with suspicion and jealousy any encroachments on their domain by "outsiders." The good college professor's understanding of the beauties and subtleties of his field, his scorn for clichés and pedestrian approaches, and his treasured intuitions about pedagogy make him a tough customer to sell on the idea that he could profit from the contributions of an instructional designer.

In his paper preceding this one, Bitzer[2] stated that in PLATOnic CAI curriculum development, "The author needs no middleman." Yet, at the University of Texas and at other CAI laboratories and centers, we have found it necessary to make instructional development a team matter through introducing a person or persons trained in instructional software engineering into the authoring loop. There is an analogue to the author-editor relationship in publishing, but the instructional designer has a more intensive interaction with the author at each stage of development than does a textbook editor and has a much greater influence on the form and structure of the resulting product. Bitzer's statement may be valid for adjunct uses, especially with his fine TUTOR language and PLATO system but becomes less so as we move along the continuum toward a mainline system, particularly at centers using languages like COURSEWRITER and less flexible graphics terminals.

INSTRUCTIONAL DESIGN PRODUCTS AND THEIR USES

It is beyond the scope of this paper to describe in detail the processes and techniques of instructional software engineering. Some relevant references are Bunderson[3] and Eraut.[4] In general, instructional software engineering has the flavor of systems engineering. That is, the context of the course to be developed in a larger system is considered; the course is considered as a "black box" with definite and measurable input and output in terms of student performance; the black box is analyzed into component black boxes; a mock-up is synthesized and tested against its output specifications; and the feedback from testing is used for revision until the system performs as specified.

A systematic procedure for instructional design must provide for management and quality control of program development. For the purposes of this paper it is sufficient to review the products of this systematic approach. These products may be classified as public documents, intermediate design products, or final program materials (see Table 1).

TABLE 1 Products of Instructional Design

Public Documentation	Intermediate Design Products	Program Materials
Brochure and Proposal	*Context Information*	
	Societal and institutional needs	
	Goals and prerequisites	Prerequisite Test
	General description of approach and justification	
	Some evaluative data	
	Production plan	
	Design Architecture and Rationale	
	Performance objectives	Criterion Test
Publications	Analysis: objectives and learning hierarchy	Prerequisite Test
Publications	Synthesis: course structure and restrictions	
	Individualizing mechanisms (flowcharts)	Diagnostic Test (if any)
	Tests to measure objectives	Criterion Test
Publications	Specification of display and response conventions; each subordinate objective	
Publications	Technical evaluation and research reports	
	Formative (revision data)	
	Summative (effectiveness and logistics data)	
	Manuscript or Author's Draft	Program (in the form of computer listing, audio-visuals, or text)
	Program steps and step formats; subroutines for production personnel	
User Manual	*Technical Documentation*	
	Program documentation for systems programmers	
Proctor Guides	Documentation for operations	
Student Guides	Student manuals	
	Production Management Plans	
	Production of all procedures listed above	

The "author with no middleman" will usually produce only the digital code and perhaps the slides, tapes, or booklets that accompany the final program. He may provide minimal technical specifications, in terms of a computer program listing. This may use characters to represent CRT characters and other non-standard graphics, and be complex and difficult to decipher in other ways as well. This minimal documentation may allow him to exchange small adjunct programs with friends or colleagues, but not main-line programs.

When people ask such an author "What does it teach?" he may be at a disadvantage without written statements of goals and objectives. If they ask "Whom does it teach?" he needs a description of the students for whom it is intended (target population) and their requisite skills. If they ask "Does it succeed?" he needs data on student performance relative to the objectives. If asked for a copy of the program, he is at a loss without user documentation and technical specifications for the programmer who will maintain it. These problems increase in severity with the size of the programs.

From the "intermediate design products" can be inferred the structure of a systematic approach to instructional development. These products consist of notes, prose passages, flowcharts, manuscripts, student data, and other ephemeral or rapidly changing forms of information. The three overall aspects of the systems engineering approach can be seen in the consideration of context (needs, goals, and justification) which result in "brochure information" useful for potential users or as part of a development proposal. Performance objectives which lead to criterion test and prerequisite test define the input-output specifications. (Other specifications in the form of constraints, such as time, may also be determined.) The analysis of objectives and definition of the system architecture in terms of hierarchy or other structure of intermediate objectives is the key step in this process. Synthesis of mechanisms for individualization and representational conventions for display and response for each subordinate objective depend on the analysis step. The special training of the instructional designer is most critical in the stage of design set off by the heavy vertical line in the Table; these are the design products that arise in connection with the synthesis of the "black box."

Formative (or developmental) evaluation implies testing and iterative revision; summative (or field test) evaluation is most relevant to the production of brochure information and professional publications—to convince others that the program works. For the empirically oriented designer formative evaluation is of the greatest interest, for it can be characterized as a continuing interaction between experiment and adjustment until the program works to satisfaction.

The main concept in the first column of Table 1 is that proper documentation for CAI programs cannot be determined until it is recognized that there are different audiences for documentation. The potential user needs Brochure Information which describes a real problem in the real institutional setting that generated the program development. The justification for using CAI to meet this problem is most crucial to the potential user. He also needs an overview of how the program works, a review of its coverage (goals) and objectives, a definition of the target population, and any validation and cost data available. Much of this same information, plus a description of societal needs and a production plan for all products, is needed by a funding agency.

Design architecture and rationale are of interest to sophisticated potential users, but full detail is most appropriate for professional publications. The pressure on universities in this country from state legislatures to concentrate on teaching the undergraduate students is in conflict with the "publish or perish" research ethic. A possible rapprochement is through doing research on the structure, organization, and pedagogical logic of one's discipline in the context of applied curriculum development projects. In some instances re-analysis of the structure of some part of a discipline for CAI implementation can reveal areas of ambiguity or contradiction which can lead to further research or more powerful conceptualizations. This is far more likely to occur in a non-physical science area, however, where theory is less quantitative. In any case professional recognition for the detailed scholarship and creativity required to develop an instructionally more economical and powerful representation of a field of knowledge should be accorded far more than at present.

Were a lone faculty author to attempt to produce a mainline CAI system by himself, he would probably fail to consider fully the "total system" aspects of the program to provide for wide dissemination. He would not be likely to consider documentation nor justification, nor have a management plan for development, nor heuristics for quality control during development.

He is most unlikely to have skills in deriving performance objectives from goals and performing a behavioral analysis of objectives. Performance objectives are the operational definitions of subsystem output which make the system testable and improvable. A well-stated objective includes both "what is given" and "what is performed," so that it leads to the specification of conventions for display and response in a rational way. Scientists have not always made the natural extension of their scientific skepticism to consideration of whether their own classroom or a CAI program really teaches what it is supposed to teach, or whether it merely selects out the brighter students.

With well-designed systems it is possible to give the lone faculty author excellent feedback from student use of a program for subsequent revision. However, summative validation studies are less likely to be produced. Research or theoretical publications arising from the Design Architecture and Rationale activities listed in Table 1 are also less likely to be produced. The lone subject-matter expert needs stimulation from those familiar with the concepts of instructional software engineering.

THE LIMITATIONS OF DISCOVERY LEARNING AND LEARNER CONTROL

Let me shift now from a logical justification for instructional software engineering to an empirical justification. This is more difficult, but the data suggest that the faculty author may need some help if he truly wishes to teach *all* of his students.

Most people I know who are in a position to work with CAI in higher education have Ph.D.'s. They are usually quite bright. In their own student days they stood out in their ability to discover answers to problems on their own and to organize, schedule, and complete their own learning activities. It is not surprising that a large majority of them take warmly to the idea of letting the student control his own learning and encouraging him to learn by discovery. It is also the case that CAI programs which provide less tutorial guidance for the multitude of possible student errors, and indeed do not consider the structure of prerequisite objectives, are far less difficult and expensive to prepare. If the target population for a CAI program consists only of highly selected, bright, inner-directed students, an author can get away with heavy emphasis on discovery and learner control. In the mass enrollment situations where CAI will have its greatest economic impact, it is doubtful that this will be the case.

A program of research supported by the U.S. Office of Naval Research and the National Science Foundation has been under way at the University of Texas for several years. This research has employed an imaginary "science" task, "The Science of Xenograde Systems." A Xenograde system consists of a nucleus containing small particles called alphons. One or more satellites may revolve around the nucleus, also containing alphons. Under certain conditions, a satellite may collide with the nucleus, exchanging alphons and affecting satellite velocity. The student must learn an algorithm to calculate the status of the system in terms of alphon count, satellite distance, and other variables as a function of time. The task has the hierarchical structure of concepts and quantitative rules characteristic of many topics in science education. In addition, its imaginary content assures us that students are totally naive as to any of the concepts at the beginning of an experiment, so that we are dealing with new learning. Perhaps the greatest advantage of its imaginary character has been to enable us to concentrate on design variables—structure, display, etc., rather than subject matter variables.

Most of the studies I will report used science education or secondary education students, primarily juniors, in the College of Education at the University of Texas. Our first study used a simulation of an imaginary device capable of recording in tabular form the states of a Xenograde system at discrete steps in time for given initial conditions. The question we sought to answer was how should one use a simulation to teach material of this sort. Various degrees of structure were provided to each of four groups of students to assist them in learning thirteen decision rules. The most structure was provided in the "Expository" group in which the rules were presented in sequence, each on a separate page of a booklet. Parameters were displayed for the student to input to the simulation so that an example of that rule might be generated. Three test items were then given which required the application of that rule. If two out of three were passed, another rule was presented; if not, another example of the same rule, followed by three more test items, was presented. The second, or "Discovery" group worked without the written rules.

The third group was the "Raw Simulation" group where students were simply told to experiment with the simulation, generating their own parameters until they understood all the quantitative relationships necessary to generate any Xenograde record, given initial conditions. A fourth, "Guided Simulation," group was given additional structure by means of test items.

The results of the study made clear what we should have known in the beginning: simulation alone is inappropriate for teaching totally new material. Students in the two simulation groups were extremely anxious, bewildered, and frustrated. By dint of prying information out of the experimenters and fellow students and perseverance, some of them did learn as well as any student in the more structured situations, but it took them much longer, and some gave up. While inappropriate for teaching new material, a simulation model may be used by a skilled instructor to illustrate complex relationships in context with much didactic instruction. A simulation may be used, after basic concepts and principles are learned, to integrate them in the context of a meaningful problem and for testing the acquisition of concepts and principles and the ability to use these in problem solving. It is also good for generating pedagogically useful examples and displays.

Our subsequent studies used only the "Expository" and "Discovery" groups. The program was revised and simplified, and the simulation aspect removed. A discovery group required significantly more examples, and hence time, to learn than the expository group which received written copy of the rules. This main difference is illustrated in Figure 1 by the X's. As is our custom, we do not like to consider mean differences without an analysis of the individual differences which are concealed therein. Figure 1 therefore shows the linear regression lines of a number of examples on reasoning ability, measured by means of separate tests, for both groups. In the expository group, reasoning ability makes no difference, but in the discovery group, it is the students low on reasoning who suffer, showing that discovery learn-

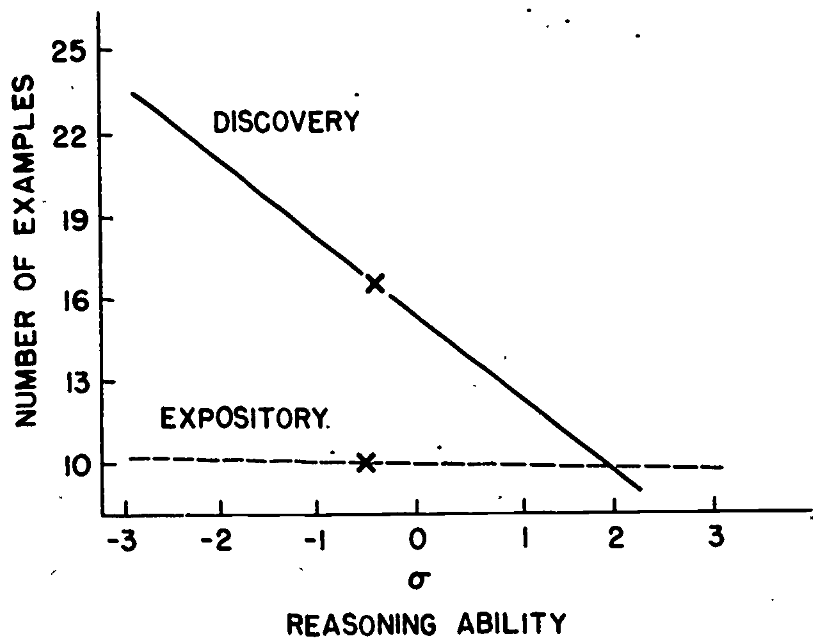


FIGURE 1 Comparison of examples required to achieve learning objectives in Expository and Discovery groups, as a function of reasoning ability.

ing, while efficient and perhaps motivating for the brighter students, places a burden on other students which an expository treatment does not. The same pattern of regression lines appeared using associative memory ability as the covariable. Students low on memory suffered in a discovery treatment but not in an expository treatment.

In spite of the greater exposure to examples in the discovery group, they did not do as well on the post-test as did the expository group. There were no significant mean differences between the two groups on a retention test taken two weeks later, nor on a transfer test that required the discovery of three new rules, given examples. This finding is contrary to the hypothesis favoring discovery learning for retention and transfer. While it is not safe to generalize too far, given the restricted nature of this task, its short duration (less than two hours) and the student population, we must recognize the real possibility that for *new learning* a carefully programmed expository treatment will be of equal or greater effectiveness and efficiency than a discovery approach, especially for the less able student.

LEARNER CONTROL STUDIES

The availability of CAI raises the prospect of learner control of the sequence of instruction. Allowing students to control what they see and do next lets them ask questions without the system designer having first to solve the natural language interpretation problem in CAI. The learner must organize his own learning, and it may be more meaningful to him to receive instruction only after he has made an active decision to ask for it. Finally, by relieving an author of the necessity of being omniscient with regard to what he should do next for a particular student at a particular time, it could greatly reduce the difficulties of program development.

Our first learner control study was concerned with the effects of learner control of sequence in new learning. A modification was made in the expository treatment to display a representation of the learning hierarchy showing the prerequisite structure of the ten rules of the simplified task; that is, which rules might best be learned before others. The student could select any of the ten lessons corresponding to the rules in any order, receiving a rule on the slide projector, an example on the CRT, and three test items. After studying these, he could select another of the 10 boxes in the hierarchy, including one studied previously. If he chose one seen before, he would see the same rule but a new example and new test items.

The experiment, a doctoral dissertation[5] in Educational Psychology by William P. Olivier,* compared the performance of students in the learner control mode with students for whom the sequence was controlled by the program. For each student in the learner control group, a "yoked partner" in the program control group was assigned the identical sequence. To increase the statistical reliability of the results, additional students were assigned randomly to the program control group with different fixed sequences. These different sequences were quantified by an index which showed the extent to which a given sequence conformed to the hierarchy. An index of 1.0 indicated strict conformance to the hierarchy, moving from bottom to top. An index of 0 represented a complete reversal, from top to bottom, while .5 represented a variety of sequences wherein as many subobjectives were taken after a higher level for which they were prerequisite as before.

The mean post-test results for the program control group as a function of the sequence index is shown by the solid line in Figure 2. You will note that as the "hierarchiness" of the sequence was degraded, learning decreased, except when the sequence was completely reversed. By covarying inductive reasoning ability, we found that students in the scrambled sequence groups who were high on this ability were not hurt. They apparently were able to infer what they had missed in the skipped lessons. Students low on this ability did poorly when the sequence was scrambled under program control.

This effect did not appear in the learner control group; in fact, an almost

*Now at the Ontario Institute for Studies in Education.

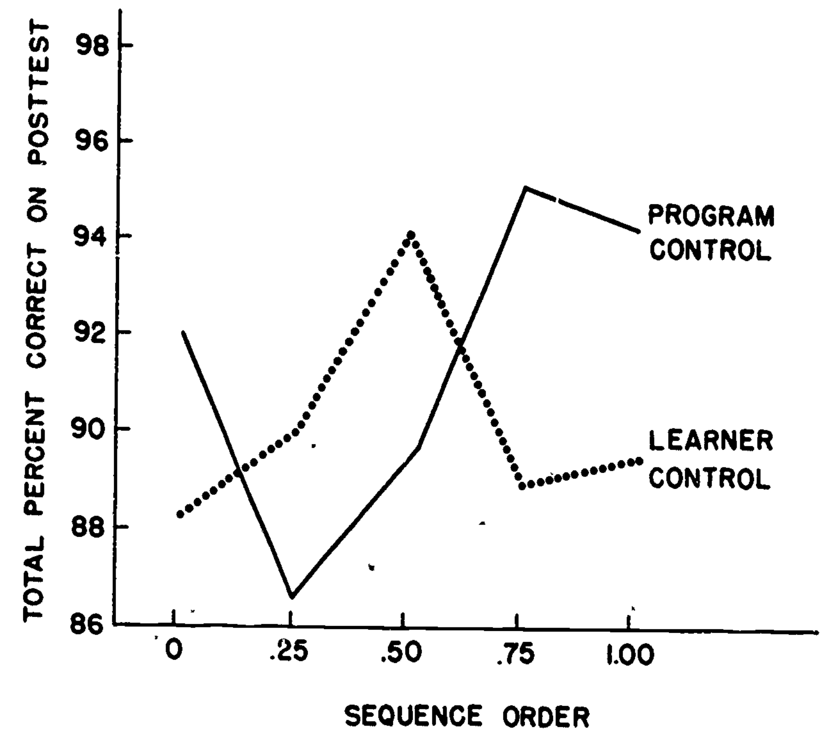


FIGURE 2 Performance as a function of sequence and control source.

completely reverse effect occurred, as indicated by the dotted line in Figure 2. Because students in the learner control group selected themselves into a sequence category while students in the program control group were assigned randomly, we know of no way to treat these data statistically as a function of the sequence index. They are suggestive only. It may be that students who are willing to grapple with the task and "explore" it by looking ahead in an idiosyncratic fashion are more highly motivated, more interested, or more creative than students who are passively willing to select a regular sequence indicated by the author. However, the program control *group* mean was significantly higher than the learner control group mean, in spite of the degrading effect of scrambling the hierarchical sequence for some.

The conclusion is very similar to the conclusion reached in the discovery learning studies: Except for a small number of exceptional students, learner control of the sequence of lessons in a hierarchy may be less effective for learning *new material* than a rationally planned, carefully designed program-controlled sequence. In subsequent studies using partially familiar material the results are less clear cut. Using a remedial mathematics program with various learner control options, Judd, Bunderson, and Bessent[6] found that in some cases program control is superior, in other cases some form of learner control. The results are so complicated by interactions with pre-test score, terminal type, amount of practice, and topic as to give the author no confidence that learner control is an easy solution to his problems of design.

SOME IMPLICATIONS

The implications of these results are clear: program development strategies that simplify the author's task by leaving important information-processing burdens with the student are likely to pose no special difficulty for the brighter students but are likely to be both less effective and less efficient for students of average and below-average ability. In this country the public funds necessary for the implementation and support of CAI in higher education are most likely to flow in the direction of programs for the economically and educationally disadvantaged, for remedial situations created by open enrollment, and for technical and vocational colleges. It is vain to believe that these students can soon learn to profit from instructional strategies developed by the elite for the elite. Instructional software engineering can succeed, for it considers by careful analysis the hierarchical structure of prerequisite objectives which must be mastered. Thus students can be started at the level of their ability and moved forward by carefully designed and thoroughly tested and revised steps to high levels of achievement. Among these high levels of achievement can be included the ability to profit from discovery approaches to learning and the ability to use learner control options wisely and efficiently.

If, by now, you accept my thesis that instructional software engineering is necessary for the full development of CAI, then important consequences follow in relation to costs and staffing. Development costs will clearly be higher when all or most of the products listed in Table 1 are developed. We have encountered costs of $10,000 per hour for the development of diagnostic and tutorial mathematics materials starting from scratch and $4700 per hour for freshman English materials adapted from an existing programmed textbook.[1] These costs included some computer and development costs which should not generalize, management costs which could be reduced by spreading them over more development projects, and programming and debugging costs which could be reduced by better CAI languages and systems to facilitate the authoring process. Nevertheless, we would be reluctant to take on a set of projects for less than $3000 to $4000 per student hour. Other estimates which may be found in the literature range from $800 to $30,000 per hour of instruction.[7,8]

Table 2 gives an example of why the costs we have estimated would be difficult to reduce further. It should be viewed more as a template with one

TABLE 2 Representative Staff and Development Costs for CAI

	One Project	Five Projects
Management		
Ph.D. (0.5 Teach, 0.5 R & D)		
One Comp. Sci., One Ed. Psych.	$19,000	→
Ph.D. – Full Time Manager – CAI	$17,000	→
Production Manager		$15,000
3 Secretarial – Business	$18,000	→
Service-Production		
2.5 Proctors	$10,800	→
2 Media Specialists	$12,000	→
2 Secretary-Keypunch-Input	$10,800	→
	$87,600	$102,600 ÷ 5 = $ 20,520
Authoring-Design		
Author (0.5 Teach, 0.5 Write)	$ 9,000	→
Consultants	$ 3,000	→
Design Editor (0.25 Time)	$ 4,000	→
2 Graduate Assistants (0.5)	$ 7,200	→
Programmer (0.5)	$ 4,800	→
	$28,000	$28,000
Total Salaries	$115,600	$48,520
Supplies and Computer Time (varies much)	$ 15,000 ?	$15,000 ?
University Overhead (50% salaries)	$ 57,800 ?	$24,260 ?
	$188,400	$87,780
Per Hour Cost for a 30-Hour Course	$ 6,280	$ 2,926
Per Hour if Course Revised and Validated a Second Year Revision-Validation (x1.5)	$ 9,420	$ 4,389

set of data than as a budget for a given college or university CAI program development center. What is needed is a set of curves representing families of budgets under variations in assumptions. Table 2 nonetheless exemplifies some of the implications of the instructional software engineering approach to staffing a modest center at a college or university. The center would provide management services to program the time-frame for completion of each product listed in Table 1, to allocate resources, and monitor progress. The center would also provide technical support services for computer operations and proctoring, especially those in support of formative evaluation; and support services for media production, printed materials, and programming consultation. Most important, instructional design services would be provided, with an "authoring team" composed of author, instructional designer, graduate assistants or other helpers, and a programmer making up the project staff.

The computer science and instructional psychology faculty who direct the center would, in their academic roles, teach graduate courses, supervise interdisciplinary doctoral students, do research, and in general expand the borders of the field of instructional software engineering.

REFERENCES

1. C. V. Bunderson, "Justifying CAI in Mainline Instruction," in *Proceedings of a Conference on Computers in the Undergraduate Curricula,* Sponsored by The University of Iowa and National Science Foundation June 1970.
2. D. L. Bitzer, R. W. Blomme, B. A. Sherwood, and P. Tenczar, "The PLATO System and Science Education," in this volume.
3. C. V. Bunderson, "The Computer and Instructional Design," in W. Holtzman (ed.), *Computer-Assisted Instruction, Testing, and Guidance,* Harper and Row, New York, 1970.
4. M. R. Eraut, "An Instructional Systems Approach to Course Development," in R. T. Filep (ed.), *Teaching Machines and Programmed Instruction, Audio-Visual Commun. Rev. 15* (1), 1967.
5. W. P. Olivier, "Program Sequence by Ability Interaction in Learning a Hierarchical Task by Computer-Assisted Instruction," unpublished doctoral dissertation, University of Texas, Austin, 1970.
6. W. A. Judd, C. V. Bunderson, and W. Bessent. *An Investigation of the Effects of Learner Control in Computer-Assisted Instruction Prerequisite Mathematics (MathS).* Technical Report No. 5, Grant GJ 509 X, National Science Foundation, Computer-Assisted Instruction Laboratory, The University of Texas at Austin, 1970.
7. F. F. Kopstein and R. Seidel, "Computer-Administered Instruction Vs. Traditionally-Administered Instruction: Economics," in R. Atkinson and H. Wilson (ed.), *Computer-Assisted Instruction: A Book of Readings.* Academic Press, New York, 1969.
8. Booz-Allen and Hamilton, *Costs of Installing and Operating Instructional Television and Computer-Assisted Instruction in the Public Schools,* Booz-Allen and Hamilton, Inc., Chicago, Illinois, 1968.

Computers in Physics and Mathmatics Instruction

KARL L. ZINN

INTRODUCTION

In this paper an overview of computer use in support of college instruction is provided with particular attention to physics and mathematics. Summary statements have been drawn from the reports of a critical study of the technology, applications, costs, effectiveness and trends for computer use in instruction. Attention has been given to the capabilities implied by information processing sciences, and to the limitations imposed by particular languages and systems. A listing of modes of computer use and some categorization of these modes are presented to organize the domain of discussion and recommendations. This listing, incorporating ten such classification schemes by various writers, may help clarify the use of these terms in technical reports and interpretations in this field.

Dimensions that may underlie various classification schemes and clarify the actual opportunities and constraints of computer use by learners are proposed. The conception of a space or domain of computer use (suggested by the dimensions) should help establish a broader perspective on computer use than that offered by a listing of modes alone. Summary guidelines are proposed for: (1) operating procedures and costs; (2) preparation of learning materials; (3) programming languages and instructional strategies; (4) introduction of computer use into instruction; (5) documentation and dissemination of findings and materials.

STATE OF COMPUTER USE IN INSTRUCTION

Many different kinds of computer and software systems are being used by instructional research and development projects in the U.S. today, and over a dozen are offered commercially. Some small machines have been programmed for use by a single student or a small number of students simultaneously;

Center for Research on Learning and Teaching, University of Michigan, Ann Arbor, Michigan.

larger systems handle up to 200 users spread throughout a city or across the country. Many of the research and development projects use general-purpose computer systems, embedding the instructional applications among other software packages available to users, and the student may use the same terminal and operating system for a variety of teaching and learning aids.

Programming languages and systems show even more diversity than the hardware. Over 50 languages and dialects have been developed specifically for programming conversational instruction. However, many of their differences are superficial, and some obvious needs of users still have not been satisfied.

The costs of using these operational or experimental systems and languages in education vary considerably. Figures reported range from $0.25 to $35.00 per hour of student use at a learning station. One reason for the wide range is the variety in assumptions made about how many effective student hours can be scheduled in a month or a year, whether the equipment is rented, leased or purchased and how much computer time will be lost to demonstration uses, preventive maintenance, or repair. The mode of student use is a big factor in cost also, but the prospect[illegible] is not likely to be fully informed about this.

Computer-based learning exercises have been written in many areas of science and mathematics and for many student levels: from remedial exercises for students deficient in high school mathematics to advanced science laboratories incorporating model building and simulation. Much of the material makes little use of essential computer contributions, and some of it would be as effective and certainly less expensive in another mode, for example, a book or film cartridge or two-person game.

In general, there is a great deal of duplication, and also much variety and rapid change. The costs of development and use of this new technology of education can be considerable, and improved communication of relevant, interpretable and current information is needed.

For example, one of the current trends is away from the programming of successive questions to be delivered by the computer under strict control of the author's sequencing rules; and yet manufacturers joining the field are repeatedly producing their own version of the "easy-to-use" author language for any classroom teacher to program just such applications. Another trend is toward generalized, even "generative" procedures which assemble curriculum material (questions and answers) as they are needed for each individual student; and yet educational technologists pursuing this course are not well acquainted with the accomplishments (and disappointments) in the related speciality of "artificial intelligence" in computer science.

MODES OF COMPUTER USE

There are many schemes for classification of modes of computer use; the four main categories which I prefer as an aid in organizing one's thinking are listed below with their typical subdivisions.

I. Instruction and the Learning Process
   A. Drill and Practice
   B. Author-Controlled Tutorial
   C. Testing and Diagnosis
   D. Dialogue Tutorial
   E. Simulation
   F. Gaming
   G. Information Retrieval and Processing
   H. Computation and Problem Solving
   I. Model Construction
   J. Demonstration and Display

II. Management of Instruction Resources and Process
   A. Student Records
   B. Materials Files
   C. Desired Outcomes

III. Preparation and Display of Materials
   A. Automatic Generation
   B. Laboratory for Development and Testing
   C. Automatic Edit and Analysis
   D. Information Structures for Representing Knowledge

IV. Other Uses
   A. Research on Instruction and Learning
   B. Educational Administration
   C. Applications in Business and Professions

These classifications are neither exhaustive nor exclusive, but are intended as a guide.

I. Instruction and the Learning Process

The most visible use of computers in instruction is to provide direct assistance for learners or trainees, or for teachers, administrators, educational technologists and others who may help or hinder learners. In any of these modes the computer-related work may be pursued individually at a terminal or in groups of various sizes, directly connected to a computer (on-line) or not connected (off-line), with real or hypothetical computers, and so forth.

II. Management of Instruction Resources and Process

Modes in this category were initially considered aids to teachers and others concerned with supervising the instructional process; however, the indirect assistance implied should also be provided to students directly without intervention of adult teachers and managers. That is, the user of a computer-based system should be able to choose between on-line and off-line assistance depending on factors such as economics and convenience. Information management services, especially those for career counseling, may be appropriately and readily extended to potential users of learning resources outside traditional educational institutions.

Within the various files for management of instructional resources and process, the essential information is about student performance, learning materials, desired outcomes, job opportunities and student interests. The user should be able to compare his own performance with that of other students summarized by goals, interests, or background. After analyzing his performance, the student should be able to move into the materials file to find suitable help and then link this self-directed activity to the job opportunity or interest file.

III. Preparation and Display of Materials

Computers can be used to assemble individualized text and problems well in advance of their use by learners; materials can then be distributed by the most economical means, although "real-time" generation of materials by the computer as needed by the student is most glamorous. The most interesting concepts in this connection are information processing procedures for handling text and graphs, and generalized or stylized techniques for producing sample exercises and demonstrations. The work of those developing educational materials and media is being changed by new technologies. Much more thinking and planning will be required; the machine will handle much of the routine, even the drafting of graphics and the editing of film.

IV. Other Uses

A number of other potentially interesting categories are listed but without elaboration since they are not central to this discussion. Nevertheless, computer users give considerable attention to support of educational administration (accounting, scheduling, planning, etc.), educational research (institutional, sociological, psychological, instructional, etc.), and the relation of instruction to various applied uses. The last area is especially important because of needs for preservice training in science, technology, management, banking, production, retailing, etc.

DIMENSIONS UNDERLYING MODES OF USE

Analysis and recommendation regarding computer use require some consideration of the dimensions that might underlie these various schemes for classification and the actual opportunities and constraints of computer use by learners. The criteria to be used for classification according to the various schemes put forth in published writing have not been clearly expressed. This is not surprising since in most cases these classifications are used only as illustrations of what might be done. It is much more interesting to determine the underlying assumptions from which categories have been derived. Some set of essential dimensions should prove to be more helpful than the total of the classification schemes.

1. Program Control

Computer-based lessons differ in the control the writer has over the student's course of study. At one extreme the student can only follow the program; typically he finds himself more restricted working at a computer terminal than he would be with a textbook or set of drill exercises. That is, the writer of a tutorial exercise has not provided the computer instructions that would permit the student to look back, review or skip ahead as he does with printed materials.

Control is desirable in some situations, and curriculum designers have used this facility of the computer to reduce inappropriate skipping about or other distractions. Control is also an advantage when the lesson designer is testing his materials and wishes to know exactly what each trial student has seen and when. At the other extreme, characterized by almost complete user control, the computer is programmed to serve the student as a tool in the management of the information necessary for solving a problem or achieving some other goal. The most common use is as a conversational computer; that is, for calculations and other immediate processing in response to directions from the student. A lesson designer can also give the student access to large files of information so he can retrieve and rearrange facts as he finds them useful for his study.

2. Diagnosis and Prescription

Another dimension of computer use concerns the extent of data recording and interpretation in the interactive program structure; i.e., the program's facility in diagnosing difficulties and prescribing new tasks on the basis of student performance and attitude. Often the computer is programmed to follow student instructions for problem solving or to present simple linear programs for practice exercises without paying any attention to the student's problem solutions or to his response to questions. At this "zero level" there is no interpretation or prescriptive feedback to the program, the student, or the author.

In typical tutorial uses the computer program processes student responses and tries to act upon them during the instruction sequence by selecting for the student appropriate remedial or new materials. Such a program could also give interpretation of response measures directly to the student, e.g., telling him why he needs more practice with or without allowing him control over future action.

Problem-solving environments, typically unconstrained, also may include interpretation of student actions, e.g., presentation of the correct form for an ambiguous instruction, suggestion of a solution procedure along a direction already taken by the student, and evaluation of his final solution for efficiency in terms of number of instructions or computer time used.

For any level of diagnosis and prescription, the extent of control over the direction of instruction may vary from nearly zero (where the student has almost complete control) to maximum control by the computer program. In other words, the program may tell the student why or on what basis he is being led through an exercise whether he is or is not permitted to change it. Similarly, any level of control allows the full range of prescription. However, if the computer is to have a justifiable role when programmed to maintain maximum control, it must be acquiring and interpreting information for use in the controlling procedure.

3. Variety of Functions Available to User

The systems dedicated to "drill and practice" are examples of a limited variety of functions available to the user: authors assemble pairs of items for presentation according to preset strategies, and students work on the exercises in sequence and within the interactive capabilities of the system. Perhaps authors may be allowed to adjust criteria for triggering a standardized computer response of "wrong, try again" or "too long, try again," and students may choose among two styles of drill exercise. Actually a drill-oriented system may allow for considerable "variety" in the sense of programming capabilities or functions believed to be worthwhile for the instructional use of a system. A package of drill exercises may include alternative ways to achieve the same performance goals, and the author may write a complicated procedure to select among them on an individual basis using observed student learning characteristics and current performance.

Although drills of limited purpose measure high on "program control," they can also measure high on the dimensions of variety and prescription. Variety is an obvious dimension to be considered in deciding the content to be utilized in a dialogue, the programming languages available for solving a problem, the alternative models to be used as a basis for a simulation, or alternative techniques available to the student to reprogram the model underlying the simulation.

4. Type of Interaction

As a dimension underlying computer use, "interaction" refers to a computer operation in which there is continuing exchange between user and program. Typically the user types some instructions, the computer types back partial results, the user modifies his original statement or continues with a new instruction, and the computer responds again, etc. "Interaction" refers not only to the speed or rate of exchange between user and machine but also to the extent of dependency of computer reply on user input and vice versa.

"Interactive" has been used as a label for certain types of programming languages and problem-oriented systems which have special features to facilitate "conversation" between user and machine. When the lesson (or system) designer knows the student (or other user) will be there to answer questions when the procedure doesn't know what to do next, or to interrupt when it begins typing out more than he finds necessary, the designer can use this situation to considerable advantage in automating only parts of the process and leaving the remainder to human interpretation and judgment. Interaction may also imply "dynamics," e.g., changes in parameters and substance to "generate" new content or rules or procedures for the student presently being aided, but in a fashion not anticipated in detail by the lesson designer.

5. Role of the Computer for the Individual Serviced

The student may see the computer as a device which does scoring, analyzing, or retrieving for him; in general, those things he would rather not do himself. Or he may see it as a test imposed on him or an exercise he must complete. If the student does see the computer as a servant doing his bidding, and a study aid doing what he asks, he may give the results of processing more attention and thereby better use the device to further his learning objectives. The teacher, on the other hand, may see the system as a proctor of required exercises and a validator of performance in testing situations. Of course, the individual students still must be proctored by some teacher or student aide if the system is to protect against one student sitting in for another or using inappropriate aids of whatever kind (books, notes, another computer, etc.).

6. "Naturalness" of Learner-System Communication

If we see this dimension as ranging from natural (conversational) to formal (precoded) language, at one extreme we find the computer programmed so that the student may respond with terms and conventions natural to him, and little if anything about system operation need be learned to use it. For language-oriented topics the computer program must be capable of performing complicated textual analysis in order to conduct a dialogue with the student. The computer's ability to process natural language, and subsequently to diagnose and prescribe on the basis of statements made by the student without any constraints, approaches the realm of "artificial intelligence" and presently is very much in a theoretical and experimental stage of development.

At the other extreme the user must learn how to give messages to the system, to recognize what has been interpreted successfully, and when to call on human aid if the system does not respond understandably. Learning computer conventions is no hardship for topics of study in which the conventions of computer use are consistent with the objectives of instruction. Formality of communication is desirable in the sciences and engineering where the computer and its programming conventions provide appropriate constraints on theorizing and model building.

Trends in Dimensions of Use

These six dimensions can be viewed as defining a space or domain of computer use, and the modes usually mentioned as simple categories (drill, tutorial, dialogue, socratic, simulation, learner-directed, etc.) are more appropriately described as filling some part of this domain. I have used this conceptualization in a tentative way to establish among users a broader perspective on computer use and to open up new possibilities for computer service to learners.

A major trend in design of computer-based exercises is a shift from program to learner control. The designer of the exercise is putting less energy into a careful diagnosis and prescription accomplished by some automated instructional strategy and more effort into providing information from which the student can derive his own diagnosis and into making available alternative interpretations and guidance from which the student can assemble his specific prescriptions.

Most systems and lesson designers are providing increasing variety of functions for the user of the learning system. More attention is being given to interaction, not simply how quickly a reply can be made to some question, but the actual responsiveness of the system to the particular input. This means that machine responses are increasingly dependent upon the commands and questions and answers typed by the student, and the lessons are designed in a way that the student is more likely to respond to information put forth by the computer.

A very important trend concerns the role of the machine from the perspective of the individual being serviced. The teacher is now more likely to see computer-managed instruction as an aid to human management than as a replacement. Probably much more important, learners find the machine more suitable as an aid to learning than as a drill master.

Naturalness of communication between learner and system is being improved day by day. Computer-based learning exercises are achieving increased relevance for the disciplines, and the nomenclature and conventions that have to be learned to use the system tend to be essential to the study of the topics (apart from the requirements of the computer as a medium of presentation).

INTERPRETATIONS AND RECOMMENDATIONS

An evaluative review of uses of computers in instruction was begun early in 1969.* A framework for discussion and review was extracted from primary sources; staff members searched for current and relevant materials in order to interpret what was being done, what had been planned, and what was judged to be most needed and important. Project directors and persons active in research and development throughout North America responded to requests for pertinent materials relating to their current work. Staff members visited sites of particular interest and attended meetings and conferences.

Throughout the project staff members coordinated their efforts with those of other projects reviewing the use of computers in education, among them: a study of the computer in higher education at Rand Corporation; a survey of computer uses in college teaching at MIT; program planning by the National Council for Educational Technology, London; and numerous single conferences and symposia. The final report of this study will be available through the ERIC Clearinghouse (see Appendix). It is arranged in distinct sections to meet the different needs of three audiences: (1) students of educational technology and interested laymen, (2) educational administrators, teachers and instructional system managers, (3) educational psychologists and computer scientists.

To the student and layman, the report offers an introduction to the present performance, problems and potential of computer uses in education. Technical terminology has been minimized in the summary and interpretive sections. Much of the specialized vocabulary for the field is clarified by common language definitions and examples in a glossary. The administrator or teacher, already familiar with the debate about computers in educational technology, will find recommended directions for computer use in education,

*Financial support from the U.S. Office of Education under Contract Number OEC-5-9-32-509-0032.

and background information on which to base project proposals and new programs. For the specialist in research and development of instructional uses of computers, materials given in support of the recommendations provide detail on differing opinions regarding computer instruction, dimensions underlying computer use in education, and comment on system design. The index should assist in location of material on specific subjects or opinion by certain writers.

The recommendations of the study may be summarized in the following set of guidelines:

Costs

Computing resources are now available at a modest cost, and the expense of computing aids will continue to decrease. It is important to select carefully among the options available, and to plan for effective use of future capabilities. People are the most important resource, and incentives should be established to encourage the participation of the best available authors, teachers, technicians and other contributors to innovative activities.

The funding of innovation should cover the time necessary to confirm the effectiveness of new resources and techniques, and to bring the best ones to fully operational status within educational systems. Cost effectiveness in the short run should not be the major consideration in a decision to introduce computers into instruction and course development programs; the benefits in the long run which follow from introducing computing and information processing now are worth some additional investment at this time.

Development and Evaluation of Computer-Based Learning Materials

A broad context should be established for review and planning of the appropriate place for computing activities within curriculum and learning activities. Determination of goals should precede selection of methods and media. The management of a computing facility for development and instruction should be oriented to its users: authors, teachers, administrators, and especially students. Effective support for users, in the form of reference materials, demonstrations, training materials, and consultation as well as programming assistance, is a major consideration in furthering efficient and effective development activity. The same requirements for user support apply to the assistance that the author should provide the teacher and student using computer-related materials. The automation of some special assistance to the user is quite promising.

Programming Languages and Instructional Strategy

No one language can be right for all users; each mode of computer use has special requirements, and different styles of program preparation benefit from different language features. Although no languages will be widely adopted as standards, translation among languages of similar purpose will be quite practical. The real needs of the learning environment should determine the procedures or information processing used; language should be shaped by the instructional uses. To remain flexible, a project should work within a general-purpose system providing more than one suitable programming language, or use a preliminary translation stage, probably automatic, to go from notations convenient for the authors into the programming language of the particular instructional computing system. The key to serving a variety of user needs is adaptability to individual requirements of authors and topics of instruction; probably this will be achieved through automatic means for extension of languages.

A curriculum development group should ultimately remain free of the constraints of specific languages even to the extent of eventually leaving the computer when the available software and programmers cannot implement the important features of an instructional strategy. On the other hand, persons or projects committed to exploitation of the computer in education should select carefully among possible applications those that best match computer contributions to the real needs of learners. Furthermore, the development of information-processing tools for learning should be pursued with a subject or discipline orientation rather than exclusively within the special area of instructional technology.

Introduction of Computers into Instruction

Determination of needs and goals should be accomplished with full participation at the local level; concerned personnel at all levels in an educational institution should guide the development of new resources for instruction which will further the goals of the institution and those it serves.

Computer-based systems should be introduced into curriculum and instruction only to the extent that they make a better contribution for meeting educational needs than alternative systems and procedures. The use of the computer by students as a tool under their control, for learning and problem solving can be recommended strongly. Management or records and materials preferably by students directly as well as through teachers and administrators, is a promising application. The presentation of instruction (exposition or remediation) via the computer may contribute significantly to the learning and favorable attitude of some students having special needs resulting from learning disabilities, cultural differences or gaps in educational background.

One practical approach to computer use in the instructional process is to make the information-processing tools and data bases directly available to the learner and let complex human skills and judgment of children and young adults be applied where scientific knowledge of the learning process is inadequate to make specific prescriptions. In general, the scholar-teacher should remain in charge of the introduction of computer technology into the teaching and learning activities, attending to the uses of the computer in his area of study, and to the advice of experts on technical matters, and most of all, to the reactions and performance of individual learners.

Transfer of Information

Each project should produce documentation that warns others about the shortcomings of an approach as well as provides encouragement to follow successful lines of development. Although the equipment used at one site may not be generally available, the successful applications can be described in procedural terms from which adaptations can be prepared for other equipment.

Sources of information and consultation should be maintained by national and regional centers, coordinated with similar centers in other countries. Conferences on computers in education should be held regularly, some of them quite broad in scope and others specific to areas of teaching and to components of the technology. Participants should include students as well as teachers in the relevant disciplines, and also potential employers and others who may be served by education and training involving computers.

National and international programs should help distribute knowledge and expertise as well as computer-based materials. Prospective personnel for a project planned at one site should work for a time at another site which is already operational. A mobile team of experts might move from place to place contributing to the effective and economical establishment of new projects at schools and colleges. A few sample programs, demonstration systems and portable terminals should be available for trial use by any institution considering establishment of a new project.

Authors should be provided greater incentives to document, distribute and revise computer-based exercises; economic and professional rewards should derive from a distribution system involving institutions, professional organizations, publishers, computing software organizations, and computer manufacturers. A first step is the early development of reviewing activities and other programs of professional societies which will increase the amount and quality of materials developed.

APPENDIX: OTHER SOURCES OF INTERPRETATION AND OPINION

The following agencies have been involved in discussions of computer-based instruction in science and mathematics and/or published materials relating to this topic:

(1) Commission on College Physics, University of Maryland, 4321 Hartwick Road, College Park, Maryland 20740;
(2) National Science Teachers Association, 1201 16th Street, N.W., Washington, D.C. 20036;
(3) Center for Research in College Instruction of Science and Mathematics, Florida State University, Tallahassee, Florida 32306;
(4) American Institute of Physics, 335 E. 45th Street, New York, New York 10017;
(5) National Council of Teachers of Mathematics, 1201 16th Street, N.W., Washington, D.C. 20036;
(6) Mathematics Association of America [MAA], 1255 Connecticut Avenue, N.W., Washington, D.C. 20036;
(7) Committee on Educational Media [of MAA], P.O. Box 2310, San Francisco, California 94126;
(8) Committee on the Undergraduate Program in Mathematics [of MAA], P.O. Box 1024, Berkeley, California 94701;
(9) Association for Computing Machinery, 1133 Avenue of the Americas, New York, New York 10036; the ACM also contains Special Interest Groups in Numerical Mathematics, Mathematical Programming, Symbolic and Algebraic Manipulation, and Computer Uses in Education.

VI

POLITICS AND PRACTICE

Introduction

In this volume much has been said about the role of the computer; but before one can have a role, one must have a computer and an operating system. Hence, in this section we present some "inside" facts on the political and economic ramifications of academic computing facilities. The first paper actually consists of three short papers discussing the relevant experiences of Lykos, Asprey, and Peckham—each one coming from a distinctly different academic environment. Appended to this paper we have also included the discussion that followed their talks. Although considerably abridged, some pains have been taken to see that the more provocative questions and answers have not been expunged.

This paper is followed by that of Miller, whose hard-headed and practical approach to the description of campus-wide computing at Stanford University should give ample food for thought. He makes the point that "smaller institutions should join forces with some of the larger and already established institutions and help in the development of regional facilities. Right now, a junior college can usually get its best buy by acquiring some terminals from a near-by university. . . ." One suspects that Peckham might disagree; perhaps, like the developing nations, some would "prefer to make our own mistakes." But if you cannot afford mistakes. . . ? A good question, and one which the new convert must decide for himself. Many large university installations are willing to oblige, but it is clearly up to the suitor to do the proposing. At any rate, do not be shy.

The Politics of Local Persuasion

PETER G. LYKOS, WINIFRED ASPREY,
and HERBERT D. PECKHAM

1. The Large-Scale Campus Computation Center

Peter G. Lykos

INTRODUCTION

The word "computer" is a misnomer; a better term would be Information Processing Machine. This is to underscore the fact that these machines are symbol manipulators, and as such will come to be integrated naturally into the communication network that already exists, such as the telephone system. At present every area of human endeavor is being impacted by the computer; there now exists a medium of communication which is bringing together people from a variety of disciplines and levels, who are addressing themselves to problems that they now find have common denominators. Thus, the social scientist is coming to grips with matrix algebra and applied statistics, because there now exist operational programs on machines which are proving very useful to him for model building and data analysis.

Graduate schools in universities are expensive. One of the justifications given for supporting a graduate school in the university is that in this fashion scholars working at the frontiers of knowledge can be attracted to the university campus, and, in this fashion, the undergraduate program would be enhanced in that the same scholars would bring their up-to-date expertise to bear on the undergraduate program. Operationally, however, what happens is that scholars working at the frontiers of knowledge do not have much opportunity for actual contact with the undergraduate program, and the new knowledge they produce is published in journals which do not address themselves primarily to undergraduate teaching, and where publication charges and

Department of Information Sciences, Illinois Institute of Technology, Chicago, Illinois. Department of Mathematics, Vassar College, Poughkeepsie, New York. Department of Physics, Gavilan College, Gilroy, California.

other constraints lead to very short and abstract descriptions of this new knowledge and its implications.

However, now that the Information Processing Machine is being increasingly used by scholars in their research, and where their work can be reduced to operational problem-solving techniques, these are being implemented in the form of computer programs and reside in the libraries of the computer memory bank of their particular university. As university computers are now coming to be interfaced to the common communications network, this means that some of the best thinking of some of the best minds, as represented by operational computer programs, can be used and accessed from remote terminals located in colleges, or in college classrooms, and in this fashion effect a transfer of new knowledge which can be represented in this form from the frontiers of knowledge to the classroom.

Administrative, or non-academic data processing, also needs to be served. It is patently obvious, however, that the rapid growth of the use of the computer in the classroom is such that, in the not too distant future, and in the majority of secondary schools and institutions of higher learning, the computer utilization will be by far the greatest in support of academic programs.

The following classification of the ways the computer is impacting higher education may be useful:

A. Management of the educational establishment including payroll, student records, staff records, registration, space inventory, general ledger, purchasing, and even test scoring, mailing lists, student guidance, and institutional modeling.

B. Education of the students.
   1. Research support such as "number crunching," simulation, control of experiments and processes, and interface to our storehouse of knowledge.
   2. Incorporation of the use of computer-based techniques as part of professional training—especially in the problem-solving and decision-making oriented disciplines.
   3. Evolution of the new discipline, Information and Computer Science.
   4. Discussion of the impact of the computer on every area of human endeavor as part of a liberal education.
   5. Use of the *C*omputer as an *A*id to *P*edagogy (or CAP, a new addition to the set CAI, CMI, etc.).

To what extent the potential of these various aspects of computer utilization has been realized varies from campus to campus.

THE PROBLEM.

The computer is transforming the ways the problem solvers and decision makers of our society are going about their business. However, the corresponding curricula in colleges and universities have not yet responded to the impact of the new techniques. A number of conditions account for this lag. University professors are busy obtaining grants and managing a graduate research program. Younger faculty members are likely to be computer users and interested in innovation, but without tenure they cannot afford to give substantial amounts of time to undergraduate programs.

University computational services do not have as a primary objective providing computer support to undergraduates. Typically, a few keypunches or typewriter terminals are available in some neutral area on campus on a first-come first-served basis, and as a result students obtain computer support on an *ad hoc* basis. At the moment, only token use is made of the computer at the undergraduate level with an essential limitation being the current hardware and software configurations.

The professional societies have not exercised adequate leadership in fostering a greater awareness of the need for computing in each discipline and of the action programs that would help to meet the need. This conference is a significant and pioneering step in this regard. There is also a lack of awareness of the several distinct ways in which the computer is coming to be used to aid the teacher in the classroom. In the lecture hall the speaker can have by his side a graphics terminal operating in conversational mode for those subjects where pictorial or graphical representation of information is important. In the laboratory or discussion section students can solve problems, sometimes writing the necessary programs, and calling on subprograms prepared for the tasks at hand. In some situations students can prepare data for already complete programs which have been prepared for them to obtain final results with minimum effort. An outstanding example is the use of business and management games. The computer may also provide all or part of the supporting system for a programmed instruction approach to some sequence of concepts. This area of use is yet in a state of infancy due to lack of understanding of the pedagogical process and the costly and sometimes inconvenient means by which teacher-authors must prepare materials for computer presentation.

Finally, there is a wall of mistrust and misunderstanding between the academic and non-academic users of data-processing services which inhibits the significant improvement in computer services which could be realized for both groups if they were to combine their resources into a single, comprehensive facility.

THE SOLUTION

Three basic requirements need to be satisfied in order that potential computer applications be discovered and implemented by the faculty and staff of a university. First, the administration, starting with the president, needs to be convinced of the potential for computers in higher education. Having arrived at a reasonable estimate of cost, the administration must make a commitment to the design and implementation of action programs for developing in-house computer-discipline awareness and sophistication. These action programs need to include employing undergraduates to function as programmer-aides to support interested faculty on a one-to-one basis.

The largest impact of the computer on the educational process is in the development of algorithmic approaches to problem solving in all the disciplines by faculty and students working together. This process is already leading to a revitalization of the teaching of all subject matter, and is constituting a basis for better definition of conceptual elements of knowledge.

Second, the computational support needs to be adapted to the user community. The process whereby this may come to pass may be facilitated by making all use of computing resources completely visible. For example, internally sponsored research use of the computer may be handled by an allocation to the dean of the graduate school and to department chairmen, using budgetary allocations which can only be charged back against the computing center accounts.

Administration of the computing facility needs to be completely transparent and responsive to the user community. Small working advisory committees made up of appropriate and representative faculty and staff should be encouraged to develop. These might be centered around the hardware configuration, the operating system, the charging algorithm, remote terminals and communications, resident programs, and files.

Third, given a smoothly running, reliable computational facility with a user-oriented management and a good interface and overlap with the user community, the prospective user must be approached with his needs in mind: What impact will this new technology have on him?

What will the computer do for him, specifically, and is there any advantage to doing it now?

If the user originates the potential computer application, is he trying to solve the wrong problem, or the wrong part of the problem?

Will the application of the new technology constitute a significant improvement? This new technology is not a panacea. Is the potential use

old technology applied to an old need;
old technology applied to a new need;
new technology applied to an old need;
new technology applied to a new need?

All these warnings can make a prospective user hesitant to begin. The following observation has proven to be helpful: "About the only thing you can do wrong is to do nothing."

2. Perils and Problems of Obtaining Computer Facilities in a Liberal Arts College

Winfred Asprey

Vassar, a four-year undergraduate liberal arts college, opened its Computer Center in February of 1967. Its "politics of local persuasion" fall historically into two eras: B.C., before the computer, and A.D., after delivery. The B.C. period which began in 1957 lasted exactly a decade; our present date is 3.5 A.D. Each period has had its perils and its problems with some vanishing, a few resolved, others still in the wait state, and new ones constantly descending.

In 1957, as chairman of the Department of Mathematics and a devotée of teaching only the purest of pure mathematics, I was urging our senior majors (all women at that time) to investigate challenging opportunities in a new field, that of computers. Had one of them asked me for details, I would have had to confess total ignorance—did you just sidle up to a computer and ask it a question? I called Dr. Grace Murray Hopper, then of Univac, a Vassar alumna and former member of the mathematics faculty.

I was so impressed by a visit with her group at Univac, that on my return to Poughkeepsie I immediately got in touch with IBM, whose property literally joins Vassar's. Subsequent conversations resulted in my accepting IBM's invitation to spend a year at their Watson Research Center in Poughkeepsie. By the end of the first month of "hands-on" experience, I was a convert; I still am.

The members of Vassar's Department of Mathematics, willing to abet my enthusiasm, proposed to the faculty Curriculum Committee that we be permitted to introduce a one-semester course in numerical analysis for advanced students. The question of the appropriateness of such a course in a liberal arts curriculum was considered so grave a matter that it was debated at a meeting of the entire faculty who finally voted approval of the course subject to critical review at the end of the term.

For several years the one-semester course, taught by a succession of outstanding mathematicians from neighboring IBM, remained the only class in the curriculum at all related to computers—now and then one or two enterprising students were given permission to develop an independent project. Then, quite unexpectedly, the New York State Department of Education became our next ally by asking me to direct an in-service institute on computers for local high school teachers of mathematics. IBM cooperated by giving us the use of a computer in their Education Building on Saturdays. A dozen *eager* students volunteered to be my assistants. "Hands-on" experience for students was at last a reality.

Shortly after Dr. Simpson became President of Vassar in 1964, he asked me to take two weeks to visit colleges with campuses similar to Vassar's to find out what they were doing, or planning to do, with computers[1]; simultaneously, he commissioned a team of efficiency experts to study Vassar's particular needs. By the end of his first year in office, the trustees had approved the *purchase* of an IBM 360.

The next move was to draft a proposal for NSF funds. Three of us: a chemist, mathematician and physicist worked excruciatingly hard to substantiate our request; President Simpson accompanied me to NSF headquarters in Washington for a consultation; the eventual award was a splendid $50,000 grant. IBM encouraged us with a $60,000 gift, the remainder of the money being provided from Vassar's general funds. Architects were hired to remodel a centrally located 65-year-old building, formerly the campus laundry. The Center took shape over the summer and throughout the fall; delivery of the 360 was scheduled for early January.

In November, the blow fell; Yale proposed marriage to Vassar, part of the contract being to move Vassar to New Haven. Groups to study the pros and cons were formed; tempers flared; students, faculty and alumnae vocalized. More serious to the on-going life of the campus was the decree to suspend all building until the issue was settled. Miraculously, two projects were in too advanced a state to be stopped, the computer and the new organ for the chapel. The Computer Center opened in time for second-semester classes, its 360 in running order.[2] The B.C. era was over. It seems long ago.

Much has happened in the 3.5 years of A.D., each development requiring different brands of "local persuasion." The single course in numerical analysis has been augmented by eight more, all listed in the college catalogue under *Computer Science Studies* and elected this year by more than 200 undergraduates, one-eighth of the student body. More courses mean more teachers; six have been authorized by the administration for next year, three of whom will be visiting staff from IBM. While students are beginning to flock to the Center, attracting faculty is a major problem. In spite of innumerable and well-attended open houses, demonstrations, short courses in FORTRAN, computer appreciation seminars and the like, scarcely a handful of faculty really use the computer, either in teaching or research. My hope is that students will act as "local persuasion" catalysts by tacitly assuming their faculty advisors are competent to suggest and evaluate computer-oriented projects.

Now, let me report on a problem of a very different nature, one that I had optimistically discarded as settled. In the B.C. era many of the humanists, predicting that Vassar would be mechanized, had opposed the establishment of a Computer Center. Students, too, feared de-individualization, being a number instead of a person. Just as opposition threatened to become a serious morale problem, everyone's attention was diverted by the engrossing demands of more critical issues; should Vassar remain "mistress of her own house," become coeducational, etc. By the time these debates were settled, the Computer Center had become part of the campus. The humanists, if they gave the question any further thought, realized that the Computer Center had little effect on either their professional or private lives. Peaceful coexistence continued until recent months.

Late in the summer of 1969, IBM gave Vassar a leadership gift of $50,000 to support exploration of the feasibility and desirability of establishing a graduate technological center in the mid-Hudson area. Dr. Charles Schaffner, vice-president of Brooklyn Polytechnic Institute, was appointed by President Simpson to head the study. Although his report has not yet been released, a vociferous group of students are actively campaigning against the center, demanding that the trustees sell all IBM stock held in the Vassar portfolio.

The humanists have joined the attack on the graduate center, the majority of them for other reasons. They have revived their earlier fears of automation; they are concerned that a graduate center would overshadow the undergraduate college, that the sciences would be stressed to the detriment of the arts, that the more able faculty would be seduced by the lure of research and the chance to teach graduate courses, that the teaching of undergraduates would be downgraded in importance, that in return for its extensive financing IBM would expect to control the educational policies of the center. Until Mr. Schaffner's recommendations are released to the members of the college community for debate, little can be argued with profit. The enforced waiting is itself a frustrating problem.

I have tried to highlight the experiences of one college with computers by emphasizing the downs as well as the ups and by picking out areas not peculiar to our own situation. One conclusion is incontrovertible: computer facilities on a campus spawn problems at a prolific rate. However, there is one problem that never arises, the problem of boredom. Precarious though it may be to live always in the midst of action, inventing and deploying "local persuasion" tactics to counter this and promote that, it is never dull.

REFERENCES

1. W. Asprey, "Computers on the Campus," *Journal of Higher Education*, *36* (December 1965).
2. "Vassar's Newest Jewel: the 360." *Vassar Alumnae Magazine*, April 1967.

3. Experiences at a Community College

Herbert D. Peckham

INTRODUCTION

This paper is directed at the practical and political problems involved in starting a computer program in a small college. The goal will be to explore a slighted but, nevertheless, a very crucial part of any serious computer program: the politics of local persuasion and its impact upon the computer program.

At the risk of restating the obvious, essential characteristics can be defined which must be present before any program can be brought to fruition. First, there must be a person or persons who are vitally interested in the program. Without this first essential, a computer program, or any project for that matter, is doomed. A second requirement is a well-thought-out plan or program which leads to the desired results. In the case of a computer program, this should be a plan that reflects equipment requirements, financing, curriculum changes, and a timetable governing these factors. It is possible, but extremely improbable that random efforts of enthusiastic personnel will produce desired results. A third and possibly too often neglected element is the catalyst known as political persuasion. The arena of political reality is the place where most probably the success or failure of a program is determined.

CASE HISTORY

Gavilan College is a small, two-year public college in the California system of Community Colleges. The present enrollment is approximately 1000. The communities that the college serves are agriculturally based and would be characterized politically as very conservative. The college is administered by a Board of Trustees elected from the District, the Board bearing the ultimate responsibility for all functions of the college. In 1965, the author was involved in planning courses and curricula in the mathematics, science, and engineering areas; a computer program was not even under consideration. Lacking any experience in computers, it was felt that any type would be much too expensive for the lean budget, would require special training, and was altogether out of the question for such a small, remote college. The thing that changed the whole outlook was the addition of a mathematics instructor to the staff. This instructor had had some marginal experience with a computer and was enthusiastic about the possibility of using the computer in mathematics instruction. After some investigation it was learned that it might indeed be possible to afford a computer.

At this point, probably the most important decision in the entire program was made. The author had seen numerous instances in which large computers were shared between administrative and educational uses, with the educational applications receiving the lower priorities. Accordingly, a political campaign was initiated to convince the college administration that the best course of action was to acquire a small computer strictly for educational use. If at some time in the future, the college had need of data processing, it would be cheaper to buy such services from commercial firms. This is in fact what transpired, leading to a happy arrangement for both faculty and administration.

Our first computer was certainly not ideal from the educational standpoint. Having a single input-output device, an IBM electric typewriter, the system was always terminal-bound. Most students required up to 30 minutes to type a program in, troubleshoot it, and get useful results out. Another problem was maintenance. Being a first-generation, vacuum-tube model, there was a continual requirement for maintenance. A temporary solution was gained by convincing one of the maintenance personnel from Control Data Corporation that his best interest lay in a college education. He subsequently enrolled at Gavilan College providing two years worth of local maintenance. This is another example of how political persuasion came to the rescue of the program.

Early in 1969, it became painfully obvious that another computer would be required for the beginning of the 1969-70 academic year. Another concentrated job of political persuasion on administration and trustees was successful, largely due to the achievements of our first program, and authority was given to lease a new computer. The computer system selected was a Hewlett-Packard 2007 educational system consisting of a 2114 computer with an 8K memory, a mark sense optical card reader, a high-speed photo tape reader, and model 33 teletypewriter was installed in August 1969. The feature of the computer system which makes extensive ulilization possible (at the present time approximately 150 students are involved) is the optical mark sense card reader. Probably 400 to 500 students could be supported with the system. Programs are prepared in BASIC and are marked on cards using an ordinary pencil. These are then input at high speed into the computer through the card reader. Thus, the only off-line equipment required is a pencil.

At the present time, the computer is used extensively in a sequence of five integrated physics-calculus courses designed for science, engineering, or mathematics majors. In addition, routine computer assignments are given in introductory physics courses, all engineering courses, and introductory statistics courses. During the next academic year, usage will expand into the balance of the mathematics offerings, as well as chemistry. Based upon the success of the new computer system, plans are under way to make use of the computer in other academic departments. It seems clear that the computer-based instruction at Gavilan College is "out of the woods" and will continue to grow. It is the firm belief of the author that the one inescapable reason for this was political persuasion, coupled with the enthusiasm of several progressive faculty members.

TARGETS AND ARGUMENTS

If the efficacy of computer-based mathematics and physics instruction is granted, then the relatively few instances of sustained computer involvement in colleges and universities is somewhat difficult to understand. There are unquestionably many reasons why the computer has not found its way into the majority of the classrooms. A possible answer to the dilemma might be a compendium of "stock" situations with suggested reactions.

Target: small college with no computer-based instruction.
Argument: we have gotten along rather nicely without computers and can continue to do so.
Counter-argument: the larger colleges and universities are moving rapidly to introduce computers into the classroom. If we fail to recognize this and fall behind, our academic program will lose credibility.

Target: large college or university with no computer-based instruction.
Argument: we have gotten along rather nicely without computers and can continue to do so.
Counter-argument: the smaller colleges, even two-year colleges, are moving rapidly to introduce computers into the classroom. We simply cannot afford to fall behind. The larger colleges and universities have an obligation to set standards of content and methodology.

Target: college or university with limited computer facilities utilized primarily for research or data processing.
Argument: we already have a computer. You are free to make use of it in your classroom.
Counter-argument: the requirements of our data-processing system are too involved to permit students to interrupt the orderly flow of information through the computer center. It would be wiser to acquire a small separate

facility which can support the purely educational utilization of computers in the classroom. In this way, we maintain the integrity of the data-processing center and, also, have unlimited student access to an educational computer system.

Target: college or university with plans for a computer center.

Argument: the educational uses of computers will be taken care of in our combined computer facility.

Counter-argument: we must be very careful to set aside sufficient computer potential for student use. Parkinson's Law is equally applicable to computer centers. If we are not very careful, student use may be excluded. In the long run, it may be wiser to have a separate computer system dedicated to student use.

Target: physics or mathematics faculty member.

Argument: I am far from convinced that computers have any place in my classroom.

Counter-argument: there are certainly many unanswered questions as to the proper place of the computer in the classroom. However, the trend is clear, as well as the implications. With the availability of the computer center, methodology and coverage become open questions in every course. To insure that we obtain full benefit from the computer without losing any of the historically established values, we need the assistance of all faculty members.

Target: Dean or President of college or university.

Argument: with NSF support drying up, and with budget situations being what they are, we simply can't afford computer-based instruction.

Counter-argument: let us make sure we are discussing the same thing. Computer-based instruction need not require the large "megabuck" facility which is so familiar. Smaller systems are coming on the market which are cheaper, need no special facilities, and lend themselves to educational use. With the current emphasis upon increasing the quality of the classroom experience, it will be easier to justify the type of computers required to support instruction. With or without outside financial support, we cannot afford to neglect this potential contribution to instruction.

The "targets," arguments, and counter-arguments above are certainly not exhaustive. They probably serve only to reveal the bias of the author toward the small computer system dedicated completely to educational uses. The point of view in this paper is a provincial one based on a limited range of experiences at a community college. Whether the specific situations are pertinent at other colleges is not the important point. The significant idea is the marshalling of ideas and plans to "sell" a program—in other words, political persuasion.

GENERAL COMMENTS AND CONCLUSIONS

There are some general techniques that can help to insure the success of computer-based instruction programs. If at all possible, it is wise to band together with other institutions to form professional groupings having the same goal. Financial assistance for such groups or consortia is unquestionably far easier to obtain. There are several regional computer centers in the United States. One of the primary goals of such centers is to assist colleges to start computer programs. These computing centers may be able to provide services at a significant reduction in prices compared with commercial rates. Finally, there are a number of dynamic programs under way involving the computer. It is important to learn from the mistakes that have been made. Those colleges or universities that have led the way constitute a most valuable resource, which one should not hesitate to draw upon.

It is always dangerous to draw conclusions from a specific situation, particularly when the person drawing those conclusions has been intimately associated with the development and goals of that program. However, it is fairly clear that any project moves from conception to reality in the political arena. To neglect the importance of this concept is not to be realistic. Simply to be right and have reason on your side is important, but it does not assure ultimate success. Success is gained in the cold, hard, light of political reality. The wise planner will keep this fact in mind from conception of his project through to its fruition.

4. Discussion

PARTICIPANTS: Arthur Luehrmann (Chairman), Peter Lykos, Harold Weinstock (for Winifred Asprey), Herbert Peckham, and Audience.

Weinstock: Of the three speakers that we've just heard, one of them is a chemist, one is a mathematician, and one is a physicist. I don't think there is any magic way in which you can designate somebody as the person to galvanize to action in order to get a computer on the campus.

Audience: I'd like to ask Dr. Lykos to discuss the use of small systems, particularly in large universities in which they are dedicated within physics departments.

Lykos: We have been doing a lot of thinking about that at ITT; the number we are dealing with is $200,000, and we are talking about a computational laboratory. It is my contention that what has been going on in this country, even in the major universities, is token academic computer use, and that we really do not have a viable mechanism whereby we can blend the computer into our general education program. The computation laboratory concept is based on a mini-computer which will support 30 terminals, the system being set up to function in several modes. One would be a higher-level language for problem solving, such as our CALCTRAN, or BASIC, or perhaps even APL. A second mode is an information-processing mode. Students in the lab could generate and edit files to be used as inputs to application programs residing on university machines elsewhere. The instructor then dials up the remote machine, sends in a batch of student inputs, and receives the processed results. In this way the computation laboratory provides a channel between the frontiers of knowledge and the classroom. The third mode would provide a CAI package whereby individual instructors can develop curricular elements of their own, so becoming aware of the problems of CAI and more receptive to other developments in it, even if it is not a large part of their own effort.

A small computer can also provide a small college with administrative data processing on campus, a need that must be served, and that constitutes an important part of the budget, and can be brought to your president's attention when you talk to him about money. But also keep in mind that commercial payroll data processing may only cost you 20¢ per check. Anyway, medium-speed printers, card readers and tape-drives are needed. There are now cassettes on the market--not to be confused with those in our little tape recorders--which are shrunk-in-the-wash tape-drives that can support two tapes, so you can do merging operations, for example. The cost of such systems is on the order of $200,000, and embodies all the elements required to bring the computer into the undergraduate program in an effective way.

Peckham: It is a good idea to consider this sort of situation, but the danger is that if you start with one idea in mind and expand it, if you are not careful you wind up with another computation center. It still seems that there is a point to be made in the small computer, designed specifically for education within the department.

Audience: In a large school the accounting scheme is a major impediment to using the computer, whether there is dummy bookkeeping or whether the department pays for educational time. Would you comment on accounting schemes?

Luehrmann: The Dartmouth scheme is singular--it's a free resource.

Au 'ience: What persuasion do you use to create a free resource?

Luehrmann: You had better ask President Kemeny about that; I am sure the answer is a book in itself. Be strong and determined.

Lykos: We have made a package available to secondary schools in the greater Chicago area, which included teacher training. In turn, we made a case to our administration for adding an interface machine which would support remote terminals, so that we could come to high school superintendents with a package costing $2000 per year: half for Illinois Bell for teletype, dataset and phone lines and half for our interface to the communication link, including enough machine time to support 50 students, each submitting three jobs per week, for the academic year. This built up pressures from outside the campus and created an image of IIT's commitment to the computer, bringing hordes of students to our campus. Last year, half our incoming freshmen in the College of Engineering and Physical Sciences had had computer programming in high school. (We offer a one-semester-hour course for incoming freshmen who are "deficient" in this regard.)

In our operation we run on a cost-back, charge-back procedure. We have a charging algorithm whereby a project account can be set up by an instructor and he can also set up subaccounts. Every student has an allocation of machine time for whatever he wants to do; I think it is an essential part of student training that he knows he is using a finite resource that costs money. He has free choice of the utility he wants to use, whether FORTRAN, SNOBOL, COBOL, etc., but he has to pay the piper. I think that this is a healthy position to have.

With charge-back procedures I can sit down at budget time with the academic vice-president and tell him exactly how much he must pay for what he wants. Another thing I have just succeeded in doing (after six years) is to make the departmental allocations exclusively for classroom use. Research allocations are administered through the dean of the graduate school.

Audience: Dr. Lykos, I feel that the publish-or-perish pressure on the faculty member causes him to avoid an administrative role as a Curriculum Development Group Leader.* The low usage of some of the IIT teaching units was due to the teachers' lack of flexibility, the fear of using units that he cannot adjust in case of classroom failure. The point is that the initial curriculum development in a discipline may call more for someone experienced in computers than it does for high-level disciplinary expertise.

*See Harold Weinstock, "Cooperative Ventures in Curriculum Development," in this volume.

Lykos: The basic problem is that of faculty training and involvement. NSF is funding a program of 15-27 month institutes whereby people would be collected by discipline and guided by a professor full-time for a summer, returning to their home institutions with some sort of common terminal or communications capability. They will reconvene the following summer to compare experiences and initiate a new discipline-oriented group into the sacred mysteries, which will then continue the work the following year. The group leader would be able to identify new promising individuals and give a good deal of guidance, enthusiasm and a sense of direction to the group.

Audience: The primary problem is not economic, it is political. People in the various disciplines engage in this work at their peril, especially those from major universities. Do you see this changing in the future?

Luehrmann: The healthiest sign is the department which itself recognizes that it would like to use computers in teaching and endorses such effort by its faculty. However, the loner in a department who wants to do computing is probably ill-advised. He ought to get out of it, unless his department is willing to declare it as viable work. There's glory to be gotten by having computer work done. The departments and institutions benefit from it. But unless a *quid pro quo* is clearly understood, it does not make sense. There are better things to do.

Lykos: I think that's the way things have been. I think they are changing. Wondrous is the influence of the Bureau of the Budget. There is increasing attention being given to the role of the computer in the various disciplines; it's even becoming a matter of survival. We are witnessing the crossing of two curves, and I prefer to regard it as an opportunity; there is a large and growing awareness of the impact of the computer on the various disciplines.

Weinstock: I think there are only two things that impress a university administration—money and prestige. So much emphasis was put on scientific research because it got them both. Nowadays, the one thing the universities do not seem to get as well as they used to, is money. On the other hand, there is a lot of money which is going untapped, because there aren't enough people asking for it in the fields of educational research, and in particular in the field of computers. I think the administrations will come around; if you are doing educational research which brings some prestige and money it will count for an appropriate number of Brownie points.

Audience: I'm from Carnegie-Mellon University, where we have enormous computer resources. It still turns out that for a lot of our purposes we can just go to commercial time-sharing services at $3-4 per hour, connect time. It also causes wonderful internal political problems when you turn down the in-house resource. All you need is a telephone number and a monthly bill. And you can get all the work that has been done at Dartmouth for nothing. What do you think?

Luehrmann: People do seem to underestimate the overhead of operating a large computer center. Especially, systems development costs.

Peckham: But you must be sure you don't get involved in a lot of line charges. If you happen to be where there is a local drop, that's fine. In my case I'd end up spending more money on the telephone than I would for computing. Also, if it is open shop, you are placed in the position of holding a club over the student's head, because it is costing money when you're on that line. That is dangerous; I want the student to feel free to use the computer. Still, it is a good way to get started.

Audience: I'd like to tell you something about the Claremont experience, because it might help the small schools. We set up the following billing algorithm: there is no charge for computer use, but each of the colleges in the Claremont cluster is billed for that fraction of the total computer budget that corresponds to its fraction of computer use in the previous year. This motivates the faculty to increase their use; they think they are getting something for nothing. That makes the center grow very rapidly.

Audience: About two years ago at Irvine, when there was much trouble about what was going to be done for computers and how we were going to run education, we had a strong group on campus who favored doing everything from outside sources. In order to get some hard figures, we sent out a very elaborate request for quotation, specifying kinds of uses, number of lines, amount of time, and so forth. We got a large number of bids, and it quickly became clear that it would cost enormously more to support the kinds of things we were doing with our local computers if we went outside campus. And it really was striking; it was going to cost at least twice as much to do it outside.

Management and Costs of Computers in Educational Institutions

WILLIAM F. MILLER

INTRODUCTION

The use of the computer in the curriculum of colleges and universities has grown to such proportions that the cost of the operations and the managerial organization within the school is a major concern to almost every institution of higher education in the country. In some cases, computation techniques are considered to be essential tools that the student must acquire in order to carry out research in or application of his discipline. In other cases, the computer is used directly as a teaching aid in augmenting classroom instruction. There are, of course, various proportions of each of these elements in any curriculum involving computation. In addition to its importance in teaching and research, computing has become an important tool for administrative and institutional service programs, for example in libraries. There are such great variations from institution to institution in educational goals, emphasis, and organizational history that no single plan can be expected to work for all. There are, however, some common problems that have common solutions.

First and foremost, I want to say that within my own experiences from the past right up to the present, "knowledge is *still* power." Sophisticated teachers, researchers, and administrators will get given jobs done more economically and more smoothly than will less well prepared ones. The success of your installations will depend very heavily upon how knowledgeable your clients are. For this reason I place great emphasis on the formal and informal teaching programs, for they are great aids to management.

Stanford has an enrollment of about 11,000 students; roughly half graduate and half undergraduate. There are seven Schools and a number of institutes and centers doing both teaching and research. In 1969 our outlay for computing was almost $8 million, including equipment, operating expenses and programming support. This does not include the salaries and related costs of the users themselves. This outlay represents about 4.6% of our total operating

---

Vice-President for Research, Stanford University, Stanford, California.

expenditures of $172 million. I shall give further breakdown of these expenditures as I come to particular activities.

THE TEACHING PROGRAMS

We have major teaching programs in all seven schools at Stanford. They are quite interdependent, with the Computer Science Department bearing the greatest responsibility for the undergraduate and beginning graduate student teaching. The course curriculum offered by the Computer Science Department comprises 16 courses intended for undergraduates or beginning graduate students and other 21 courses for graduate students. This department handles about 2000 students per year, aside from graduate students in the department, and most of them are taking introductory courses.

In addition to these, the Computation Center offers about a dozen "quickie" non-credit courses which vary from a 4½-hour course in BASIC to a 16- to 20-hour course on OS/360. The Computation Center processes about 2000 students per year, mostly beginning graduate students or research associates who want to get on the machine quickly for use in research. Between them, the Computer Science Department and the Computation Center process about 4000 *new* users per year, which gives some idea of the educational effort needed to support a major university computing operation.

We have found that well-prepared introductory courses cost less per student than ill-prepared ones, and the variations are surprisingly large. We have had well-prepared three-hour courses cost as little as $20 per student, while others have gone as high as $60 per student. Now we are running an average of about $30 per student for the introductory courses. For $30 a student gets a great deal of computing experience. Averaged over all courses offered by the Computer Science Department, it costs about $43 per student per three (quarter) hour course. This includes the compiler writing, artificial intelligence, and operating systems courses which require substantial amounts of computing time per student.

We are just completing our second year of a very active computer orientation for our two-year Masters of Business Administration (MBA) program. Each new MBA student is required to take an introductory computer course which includes a laboratory using our interactive BASIC system. The laboratory is expensive, about $75 per student. At the end of the laboratory course the students are running some very large mathematical computations which consume many machine cycles. In the ensuing quarters, the MBA students are given about $25 per quarter for computing. This leads to an average over the six quarters of the program of about $33 per student per quarter. We now have a serious effort under way to bring this cost down, and in the coming year we expect to run the program at a somewhat lower cost.

The report of the President's Science Advisory Committee entitled "Computers in Higher Education," February 1967, known as the Pierce Report, estimates a need of $60 per undergraduate student per year for undergraduate computer instruction, for all undergraduates enrolled at the institution (not per student enrolled in a computer course). We are spending considerably less than that amount at Stanford. We allocated $475,200 for all classroom teaching, graduate and undergraduate, for school year 1969-70, or $43 per student per year. Of this allocation, about $120,000 is for undergraduate education. This amounts to about $22 per undergraduate (total) per year. With careful preparation, we can reduce these costs a small amount, but I suspect we are close to our lower limit.

Let me now complete the picture by describing the rest of our graduate programs and the Stanford University network program. We have Masters and Ph.D. granting programs in four Departments in three different Schools and in one Institute which prepare professionals in various aspects of computer sciences. In addition to the Computer Science Department, these are the Communications Department (information retrieval, large data base systems, library systems, and communications), Electrical Engineering Department (switching theory, logic design and computer organization), Graduate School of Business (simulation and computer systems modeling), and the Institute of Mathematical Studies in the Social Sciences (computer-aided instruction).

A recently announced joint program between Electrical Engineering and Computer Science leads to a "Master of Science in Computer Science: Computer Engineering." This is a terminal program for those students who develop a competence in the design of substantial hardware/software computer systems. It is intended to help meet the very large demand for practitioners, as opposed to researchers.

There is one more program to mention; that is the Network Program in the Stanford Computation Center, serving four Bay Area colleges (Hayward, Mills, San Francisco State, and University of San Francisco) and one high school (Gunn). The program provides the full spectrum of Stanford Computation Center services that are available on our terminals, as well as remote batch card reader/printer capability.

COORDINATION AND ALLOCATION OF RESOURCES

I have described a large number of computing programs spread throughout the University. A key question is how do we keep them coordinated. In March 1968, I was appointed Associate Provost for Computing, responsible for coordinating the academic programs; I had a line responsibility for the Stanford Computation Center, and a staff responsibility for the Institute for Mathematical Studies in the Social Sciences. That was a part-time job which I did in addition to my teaching and research. When I assumed my current post as Vice President for Research, I included the position of Associate Provost for Computing. I consider this a satisfactory arrangement only because of my experience in computers, otherwise I would consider it essential to have an Associate Provost for Computing or some such position. In the capacity of Associate Provost for Computing, I act as an interdisciplinary Dean, ensuring that the departmental committees plan their curricula together. I also allocate the University's computing funds, so that Schools or Departments that wish to initiate new programs that require any significant new funding must talk to me first and I can make sure that they are coordinating their efforts with established programs.

The funds for teaching and unsponsored research (including thesis research and research by new faculty) are allocated to the Schools through the regular budgeting process. The Associate Dean or other delegate of the School discusses his needs and programs with me. These funds are not transferable to other uses. This process has made the Deans much more aware of the costs of computing. We have a breakdown of our costs by department and course, which we use as a basis for estimating costs of introducing new courses.

Our allocation of University money by Schools is shown in Table 1. These funds do not include any federal support money and constitute about 31% of the $2.5 million yearly operating budget for our campus computing facility.

TABLE 1 School Year 1969-70 Allocation for Computing

School	Instruction	Unsponsored Research	Sum	Percentage of Total
Humanities and Science	$217,700	$ 75,300	$293,000	38.0
Engineering	115,000	115,000	230,000	30.0
Business	105,000	25,000	130,000	17.0
Earth Sciences	20,000	20,000	40,000	5.2
Education	4,000	18,000	22,000	2.7
Medicine	4,000	12,600	16,600	2.1
Law	3,000	3,000	6,000	0.8
Food Research Institute	4,500	13,500	18,000	2.3
Other	2,000	12,400	14,400	1.9
TOTALS	$475,200	$294,800	$770,000	100.0

THE RESEARCH AND SERVICE PROGRAMS

In addition to the shared Computation Center facilities, we have at Stanford a large number of dedicated facilities concerned with data acquisition and control. They require substantial amounts of dedicated time for systems development. For example, the facility in the Artificial Intelligence Project is concerned with data acquisition from a camera and a microphone, and the control of the camera and the mechanical hand. The work on this project requires a great deal of dedicated systems programming effort, and could not be easily carried on in conjunction with the operation of a service utility. We have other such examples in the Medical School, Hospital, the Stanford Linear Accelerator Center, and the Institute for Mathematical Studies in the Social Sciences.

The Computation Center operates three facilities: the Campus Facility, the SLAC (Stanford Linear Accelerator Center) Facility, and the ACME (Medical Facility). The Directorate of the Stanford Computation Center is responsible for staffing, budgeting, and technical coordination of the activities of these three centers. Each of these facilities has a Facility Director responsible for the operation, maintenance and development of projects for that facility. In addition, we have common development projects that span the activities of all three facilities. In addition, the Computation Center is engaged in a joint project with the Library for the development of a library automation and bibliographic search capability.

The Campus Facility operates an IBM 360/67, SLAC, an IBM 360/91, ACME, an IBM 360/50, and the Administrative Data Processing Facility operates an IBM 360/40. We have selected an IBM 370/155 for the combined Administrative Data Processing and Library On-line Data Facility to replace the IBM 360/40.

The use of the computer for the administrative functions of the university is a very important utilization. The traditional uses in the Controller's Office in inventory control, in student services, and other record-keeping services of the university requires a very competent staff and competent management. The uninitiated, and even the initiated, can easily get snared into very extensive and expensive developmental projects. Even though the techniques of file management and data management are well known, the adaptation of these to a particular institution can be quite extensive. These developments need to be carried out under rigid cost control and careful planning.

To give you some idea of the cost of these institutional service programs, I can cite our new Administrative Data Processing Facility, which will have exceeded $700,000 in development costs. Our Library On-line Data Facility with inventory control of the books, a facility for bibliographic search for limited collection, and a facility for the behavioral science data base and an ecological data base will have cost in excess of a million dollars for development. The continuing operation of the combined facility, after development, will be in excess of $700,000 per year. The development costs were sponsored by the Ford Foundation, the Office of Education, and the National Science Foundation, respectively, but the on-going operational costs will necessarily be borne by the institution. Clearly, not every institution can afford to engage in such activities; we have to be in the position of sharing the software of these developments.

THE DEVELOPMENT OF A FACILITY

The development of a hardware/software facility for utilities and service, as distinct from dedicated research facilities, should proceed very much like the building of a house. One can view the development in five stages:

1. Decide who is going to live there—who are the users and what do they wish to accomplish.
2. Decide on the size of the facility—how many users, the frequency of use, the size of the data bases, etc.
3. Determine the rough layout—the principal modules of the system. Decide on changes, if any.
4. The blueprints—select the language, algorithms, data and file structures, data format, hardware configurations, etc. Changes at this stage are difficult and costly.
5. Implementation—converting the blueprints into an operational system; this includes coding, testing and debugging, development of diagnostics, and establishment of the operating procedures.

It is important to carry out frequent management reviews of the progress of a development and, in addition, it is very useful to have an outside consultant examine the project. A continuing frank appraisal by an outside consultant is quite beneficial to both the management and the technical personnel responsible for the project.

Developmental costs are very high. An institution that does not have experienced technical personnel and management should be very wary of engaging in any substantial financial development. There is, of course, educational value in such developments but that should be fairly recognized as part of what one is buying when an institution undertakes such an enterprise.

I believe that smaller institutions should join forces with some of the larger and already established institutions and help in the development of regional facilities. Right now a junior college can usually get its best buy by acquiring some terminals from a near-by university or college which has developed a time-sharing or remote-entry system.

There are a number of features that are important considerations in the cost of installations:

1. Multiple processors cost more money; that is, the more language processors and special systems one has, the greater the cost. It costs more to run the facility and it costs more to develop and maintain the facility. An institution may well decide that these costs are worthwhile, but that should be a clearly recognized and carefully considered decision.
2. Specialized scheduling. Any specialized scheduling either explicitly or implicitly imposed will be costly to the installation. Specialized scheduling may be explicitly imposed by manual intervention and setting up of blocks of time for various activities, whereas implicit specialized scheduling may occur by permitting the user to develop special control over resources which are then blocked from availability for other users.
3. Diagnostic and measurement tools. These tools are quite important for analyzing programs both for linguistic and dynamic features. It is important that these measurement tools identify unusual requirements of the programs, such as excessively long readings and writings so that the user may optimize them and, in general, provide the maximum information possible about the running of these programs.

In closing, I should like to say that computers are indeed a powerful tool in the education and research fields and will, in addition, make substantial contributions to administration and institutional service programs. We do need to develop a more refined understanding of the costs of particular kinds of uses and to develop both the talent and information to permit good management of these facilities.

VII

THE FUTURE

There is a history in all men's lives,
Figuring the nature of the times deceas'd,
The which observ'd, a man may prophesy,
With a near aim, of the main chance of things
As yet not come to life, which in their seeds
And weak beginnings lie intreasured.

——— William Shakespeare

How to Make Dreams Come True

ANTHONY G. OETTINGER

That computers have a future in science education is no longer seriously in question. Many older scientists still have doubts but, as a younger generation increasingly literate in computing advances through the ranks in all walks of life the need to use computers and the means to do so will increasingly be taken for granted. The case for this future has been made elsewhere[1,2], however, the details and the time scale of this progression are still bound up in complex questions of science, technology, economics and distribution. This paper deals with some of these questions.

Just as the automobile was first seen as a horseless carriage, so the computer was initially conceived of only as a tool for automating the mathematical methods that preceded its invention. As students who have learned FORTRAN, BASIC, or the like in high school increasingly come into contact with instructors accustomed to using computers with the same ease as oscilloscopes or slide rules, instruction stands to gain much in depth. For example, one of my colleagues recently inspired three juniors in an undergraduate seminar to model Fermi's original atomic pile using a conventional computing service now available at teletype terminals in their dormitories. The thrill these students had in watching their simulated pile go critical at precisely the predicted value was matched only by the instructor's pleasure in the concrete evidence of effective understanding of the underlying physics, engineering and mathematics that the students demonstrated in the process. Everyone has similar favorite examples. My aim is not to catalogue these but rather to discern the conditions that make for success.

Educational applications of computers have lagged behind commercial, industrial, military and research applications. The bridge between the early experiment that raises high hopes and the widespread realization of those hopes is built by major capital investments and by the often painful evolution of supporting institutional arrangements. The development of reliable computer hardware and software required major investments. Characteristically, such developments have been aimed at markets other than or broader than

Aiken Computation Center, Harvard University, Cambridge, Massachusetts. Work supported in part by the National Science Foundation under Grant NSF-GY-6181 and by the Bell Telephone Laboratories under a contract with Harvard University.

the educational market. Consequently, the products have required special adaptations ranging from accounting routines that deal with the large number of small jobs, typical of much of current educational use of computers, through the development of specialized compilers stressing speed of compilation more than efficiency of object code.

The economical distribution of computer language manuals and other learning materials depends on the marketing and distribution networks established by the computer manufacturers and by the publishing industry. The very existence and operation on a campus of a computer facility capable of supporting student use is itself the outcome of years of local politicking and persuasion.[3] The right of the students in an undergraduate seminar to have access to this computing system is a consequence of national and local policy decisions leading to the acceptance of the cost as a legitimate component of instructional budgets.

The pile simulation exercise mentioned above took place in a special projects seminar and not in a regular course for tens or hundreds of students. At present, multiplying such ambitious student projects by the hundreds would overtax the financial resources of most schools. Moreover, there are still few instructors capable of originating and supervising such activities and few well-developed and widely distributed manuals, textbooks, programmed sequences or the like that would otherwise guide students along such paths. Non-trivial use of computers even in relatively tried and true applications entails investments and institutional arrangements that are still ahead of us.

For instance, the widespread introduction of computer-based calculus courses[4,5] entails a far greater investment in human and material resources than do occasional seminar exercises. The success of such ventures obviously depends on solving the developmental and logistical problems of constructing *coherent* new curricula. It also depends on the much more difficult job of proving to colleagues in mathematics departments and to educational administrators that these new approaches are as good as or better than the present conventional courses. The wry title "First Things Last" of this morning's session suggests the importance of paying earlier and greater attention to the politics of local persuasion.

By far the hardest part of the task of local and, indeed, national persuasion lies ahead. The prevalent educational uses of computers in colleges have ridden the intellectual and economic coattails of research. Those few cases where it is otherwise belong to the dream, not the reality. The cutbacks in funding for research have made imperative an independent evaluation of the case for computers in undergraduate education.

At the frontiers of science the computer is now reshaping the whole cycle of scientific thought from constructing, interpreting and evaluating experiments to constructing, interpreting and evaluating theories and back again. As the automobile has transformed our society and as earlier new instruments have broadened the scope of scientific inquiry and altered its character, so it is with computers in science. The advance of science has been marked by a progressive and rapidly accelerating separation of observable phenomena from both common sensory experience and theoretically supported intuition. Anyone can make at least a qualitative comparison of the forces required to break a match stick and a steel bar. Comparing the forces needed to ionize a hydrogen atom with the force that binds the hydrogen nucleus together is much more indirect, because the chain from phenomena to observation to interpretation is much longer.

It is by restoring the immediacy of sensory experience and by sharpening intuition that computers are reshaping experimental analysis. In addition, a physical theory expressed in the static language of mathematics often becomes dynamic when it is rewritten as a computer program; one can explore its inner structure, confront it with experimental data and interpret its implications much more easily than when it is in static form. In disciplines where mathematics is not the prevailing mode of expression, the language of computer programs serves increasingly as the language of science. The simulation of Fermi's pile in a seminar is but a forerunner of the introduction of such an outlook into the more routine aspects of instruction. That is why courses conceived in this spirit represent such a revolutionary break with present practice and will require such profound institutional transformations to accommodate them. Nothing less is at stake than the healing of the 100-year-old break between pure and applied mathematics.

The technology of computers themselves is still undergoing rapid transformation. New and increasingly more convenient programming languages and operating systems are being developed. As time goes on, FORTRAN and its relatives may increasingly be seen as liabilities committing users to the horseless carriage viewpoint. The development of graphical techniques promises to relegate the printer or the teletype to the realm of the goose-quill pen. The ability to present dynamically not only alphanumeric information but arbitrary two- or higher- dimensional visual images seems likely to revolutionize our basic means of scientific communication. To those who have already had the experience, the ability to view mathematical functions with ease as curves or surfaces varying, if necessary, through time means more than a convenient synoptic representation of a table of numbers. Such experiences presage a profound alteration in our intuitive grasp of mathematical and statistical relationships.

FORTRAN and the teletype instrument have achieved longevity, reliability and ubiquity. For graphical languages and graphical terminals to be transformed from laboratory tools to everyday instruments will inevitably take time, in spite of brave words about accelerating rates of change. More fundamentally, computer graphics in education is currently in a horseless carriage state, with people using graphical terminals to make illustrations, slides, videotapes, or films. The merit of such transitional applications is undeniable. But the film, for example, is unsatisfactory as an educational medium in many ways that computers need not share.

Film technology today is very inflexible. There is very little alternative to running a film from beginning to end, precisely as it was made. The kind of random personal selection, rapid editing, variable speed of browsing, or prolonged contemplation that are so easy with book technology, particularly since the advent of dry copying, are impossible with film technology. Film editing requires either professional services with inherent high costs or delays, or else elaborate facilities that are rarely available to an instructor and which, in any case, he would find difficult to use without much training and patience. The status of property rights in visual materials is both restrictive and chaotic. The art of filing, indexing, retrieving and distributing visual educational materials is in a state so primitive as to make the state of the technical literature seem idyllic. Hence, even if we were to be successful, technically and economically, in grafting computer graphics to film technology, the achievement might be of dubious value.

I am much more sanguine about the use of videotape as an experimental visual medium in spite of the lower resolution and poorer graphical quality of this medium. It is much easier to experiment with videotape production than with film production, although the process is not free from difficulties. The electronic technology of videotape and related media suggests designs for semiautomated indexing and editing facilities that would be much more difficult and expensive to achieve with existing film media. Both film and videotape technology are excessively oriented toward production for the mass market. Videotape offers the greater hope of effective adaptation to demands for simplicity and flexibility in production and editing.

We have also experimented with the tape recording of the relatively narrow bandwidth signals traveling between a computer and a graphical terminal such as those of THE BRAIN.[6] The cathode-ray-tube terminal of THE BRAIN lends itself to playing back either material recorded from the intermediate telephone line or material originating directly from a computer. This makes it possible to intersperse canned materials with live exploratory interactions. The "snapshot" facility of THE BRAIN system enables us to store the precise state of the computer at the time a particular image is recorded. When playing back the recording, playback can be interrupted, the connection to the computer restored, the snapshot loaded back, and the computer made available for guided or free-wheeling visual explorations prior to resuming the recorded sequence. Such techniques offer the hope of optimizing mixes of the very inexpensive but inflexible audio tape with the much more expensive but also much more flexible on-line computer.

As one wrestles with development problems in computer graphics, further problems emerge. The development of graphical control languages of adequate power and satisfactory ease of use is as difficult as it has proved to be in every other realm of computer application. We have also begun to encounter problems in human perception. When a graphical terminal is used on-line for free-wheeling exploration or guided laboratory work a student can set his rate to suit his absorption and comprehension rates. However, in our attempts to use these devices as animated blackboards in the classroom, we have run into two kinds of serious temporal difficulties. On the one hand, boredom and loss of contact while the instructor was wrestling with the graphical control language slowed the proceedings down to an unacceptable rate[6]; on the other hand, during periods when the flow of graphical images came easily, there were clear indications of information overload for the students.

I am confident that satisfactory technical solutions can be developed for the first problem, but the second problem may be more subtle. It therefore seems important to explore this saturation of the students by information coming at them at an excessively rapid rate. We need to know how much of this might be attributable to fundamental limitations in visual channel capacity and how much to years of training in the decipherment of static printed verbal or graphical information without any training in two-dimensional dynamic visual perception other than that afforded by the movies or commercial television.

George Miller,[7] discussing in information-theoretic terms the limits on absolute judgments of stimuli and on the span of immediate memory, pointed to "severe limitations on the amount of information that we are able to receive, process and remember." He also suggested that the enormous discrepancy between the limitations revealed by experiments and our much more powerful performance with ordinary language might be attributable to recoding of information. "Our language," he wrote, "is tremendously useful for repackaging material into a few chunks rich in information. I suspect that imagery is a form of recoding, too, but images seem much harder to get at operationally and to study experimentally than the more symbolic kinds of recoding." The tantalizing possibility of using dynamic images as a direct communication medium is a strong incentive to the opening of a fresh attack on this problem. Making the dream of computer graphics as an effective aid to learning come true therefore entails deeper exploration both of technology and of fundamental problems in the psychology of perception.

Not only is hardware undergoing profound alterations, but so is the technology of the educational process itself. Early emphasis on computer-aided instruction—narrowly conceived as a way to automate more "classical" programmed instruction—is rapidly giving way to much broader conceptions of the use of computers as instructional aids. The array of possibilities is still bewildering. Rigid tutor or drillmaster, instrument for discovery or for freely motivated numeric or non-numeric calculation, maker of slides or movies, instrument for simulations and games, animated blackboard in the classroom, private tool in the office or residence hall, batch processing or interactive, the computer presents many opportunities and problems.

Given the complexities of making the most of computers, it is surprising that those concerned with computers in education have tended to ignore other media. Only one paper in this whole conference has a title expressing explicit concern for multiple media and for comparing their relative effectiveness. Conventional chalk-and-blackboard, book technology, and the various direct or canned forms of visual communications surely remain conceivable alternatives to computer usage in many of the realms where claims are made for the efficacy of computers.

In some instances, such as the preparation of programmed instruction sequences, the computer might be most effective as the tool for preparing the materials and experimenting with them, while more conventional media might be more appropriate vehicles for widespread distribution of the product. Repeated testing of variant programmed instruction sequences with large samples of students, may be a situation in which the flexibility of the computer and its ability to help in interpreting experimental results can justify its cost.

How much more effective than an equivalent printed program a computer might be in the widespread application of programmed instruction sequences is a question that I think has not received sufficient attention. Anyone who has watched a clacking teletype printing long paragraphs of textual material during some instructional sequence cannot fail to be impressed by the absurdity of the process. Surely the binding of these messages into a printed manual is a reasonable alternative worthy of consideration. The computer might then serve to give, at most, terse references to pages to be read in the manual and act primarily as the tool for calculations or other procedures impossible for unaided man to perform.

The argument that having material in a computer makes change and evolution easy only has merit in an experimental situation; rapid change and evolution are incompatible with widespread routine use. When, as some hope, hundreds of teachers and thousands of students are to make use of one common system, continuing rapid changes in either the form or the content of the system can produce only chaos, as is well known to the users of developmental systems! Once the necessity of freezing designs to enable widespread distribution of a product is accepted along with the need for discrete updating, as in semiannual or annual editions, the potential economic and logistical advantages of conventional print become evident once more.

While the unit cost of computing continues to decline, it remains expensive by academic standards; the more so since it tends to displace invisible costs or create new benefits whose dollar value is difficult to assess, while incurring costs that are quite easily if not always intelligently accountable. Emphasis on cost and accountability is likely to increase rather than decrease in the future. President Nixon's message on education reform[8] charged a proposed National Institute of Education with taking the lead in developing new measurements of educational output and means for applying such measurements toward greater emphasis on accountability and productivity. The message stressed that "School administrators and school teachers alike are responsible for their performance, and it is in their interest as well as in the interest of their pupils that they be held accountable." It further stated that "in opposing some mythical threat of 'national standards' what we have too often been doing is avoiding accountability for our own local performance. We have, as a nation, too long avoided thinking of the *productivity* of schools."

The realm of the quantifiable is still a small part of many aspects of the educational process, hence the pressures toward more effective and more precise measurements will, more than likely, lead to fresh excesses of numerology. But, however much we might hate to admit it, much instruction, especially in the elementary aspects of various college subjects is just mass production. If we are willing to admit that mass production is an element of college education, then to that extent the mass production capabilities of computers and related technologies might be used far more effectively than at present, when faddish emphasis on individualization, flexibility and adaptability of content to individual students succeeds only in highlighting the advantages of printing and book technology.

It seems necessary also to study the costs of conventional ways of doing things with greater care then at present. Accounting for its own operations is one of the most mismanaged facets of university administration. If the pace of the introduction of computer use in education is to be accelerated, far more attention to the development of solid comparative cost figures will be necessary than is presently the case. The question will eventually disappear as a new generation accustomed to computers becomes preponderant. As the railroads, automobiles, and airplanes replaced the horse and buggy and as electrical power and light superseded the horse and the candle, the national bill for transportation and for power went way up, not way down. If, as seems to be the case, computing is to grow as these other technologies have, the bill for computing will tend to be accepted just like that for transportation, power, and light. Future debates may therefore be expected to rest not so much on questions of dollar cost as on broader social questions arising from the transformations that computers induce in our society.

The distribution of computer services will also have a profound effect on how computers are used in science education. The traditional choice between the small but local computer and the large but remote and often inaccessible computing center is giving way to a broader range of choices typified by remote batch terminals and an increasingly wide range of remote interactive services. The failure of the telephone network to plan ahead for data transmission and its negative reaction to this challenge by raising its rates will inevitably delay broader distribution of computer services through telecommunications channels. Nonetheless, the emergence of satellite and microwave long-haul capabilities, coupled with the rapid development of cable and other local distributions systems may, through creating active competition for the telephone companies, produce a profound alteration in the technical and economical picture in this realm. Those concerned with the development of instructional technology cannot fail therefore to be concerned with the development of data communications, a field now undergoing profound technical, economic and regulatory turmoil and transformation.

In June of 1968 the FCC ordered the removal of certain tariff prohibitions against interconnection and foreign attachments. The consequences of this decision (the so-called Carterfone decision) for educational use of computers and communications are likely to be profound.[9] Without prohibitions against interconnection the potential economic advantages of private internal networks on university campuses become realizable. Moreover, there is a strong possibility of competition for the telephone companies in both the long-haul and the local distribution of telecommunications. Those concerned with educational technology need therefore pay the closest attention to the efforts of organizations like Microwave Communications, Inc., and University Computing Company (through its subsidary Data Transmission Company) to offer long-haul data transmission services. In February of 1970 the FCC issued a rule prohibiting telephone companies from entering the cable television business and also requiring them to make line attachment rights available on a non-discriminatory basis to CATV companies. This opens the possibility of competition in local distribution.

The technical possibilities of cheap, broadband digital transmission have been well understood by the common carriers and their emerging competitors for some time. How much competition will materialize in telecommunications and to what extent it will indeed prove beneficial to educational applications remains to be seen. The last word has not yet been said in the welter of commercial, legal and regulatory actions recently taken or now being contemplated. We can, however, explore some of the consequences of the assumption of sharply decreased cost of more abundant telecommunications. They are best appreciated if one thinks in terms of effective office or classroom sizes in the thousands of square miles rather than the hundreds of square feet.

The college or university as we now know it will find itself more closely linked to neighboring institutions. Indeed, it will be in greater competition not only with those institutions but with numerous other existing or emerging agents of education whose physical location will make increasingly little difference. This will be a critical factor if, as some estimates have it, the number of houses wired for CATV grows from the present 8% to 25% in 1975 and 85% in 1985. I think that the shape of the new telecommunications and of its institutional implications will, as was the case with computer technology, be determined primarily by commercial, industrial and military applications rather than by educational ones.

Nevertheless, the colleges and universities will bear an important responsibility for maximizing the potentialities of the new technologies and their institutional realizations for education. The pessimism I expressed in *Run, Computer, Run*[10] was based on my observations of the rigidities of elementary and secondary educational institutions which make it impossible for these institutions to react to change. Universities are far from being free from rigidities. Like all of the educational enterprise they are now shackled by rising costs and static or diminishing incomes.

But there is a far greater opportunity for free experimentation and significant innovation in the universities than there is in the schools. Moreover, this is where the teachers, the administrators, and indeed, the leaders in commerce, industry and government will come from. If we improve our own practices and practice what we preach, we have the power to make some of the dreams described in this conference come true. Physician, heal thyself!

REFERENCES

1. A.G. Oettinger, "The Uses of Computers in Science," *Scientific American*, September 1966; reprinted in reference 10 below.
2. I. Taviss (ed.), *The Computer Impact*, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.
3. W. Asprey, P. G. Lykos, and H. D. Peckham, "The Politics of Local Persuasion," in this volume.
4. H. M. Schey, "The Use of Computers in Teaching Calculus," in this volume.
5. W. B. Stenberg, "A Computer-Based Calculus Course," in this volume.
6. R. DesMaisons, R. Dirkman, and M. Hepner, "THE BRAIN System," in this volume.

7. G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," in *The Psychology of Communication: Seven Essays*, Basic Books, New York, 1967.
8. *Message on Education Reform, Office of the White House Press Secretary*, March 3, 1970.
9. *Addressed Cable Delivery*, Information Industry Association, IIA Response to the FCC Docket 18397, Part V, December 19, 1969.
10. A. G. Oettinger, *Run, Computer, Run: The Mythology of Educational Innovation*, Harvard University Press, Cambridge, Mass., 1969.

Higher Education and the Post-Industrial Society

ANDREW R. MOLNAR

THE POST-INDUSTRIAL SOCIETY

It has been observed by Tyler[1] that, in 1860, 80% of our labor force was engaged in producing and distributing goods, while only 20% were providing for our non-material needs. Last year, he notes, only 40% of the work force was involved in the production and distribution of material goods while 60% was furnishing our *non*-material services. The United States, says Tyler, is the first nation to have developed technology to a point where less than half the labor force is required to furnish material goods, and now less than 10% of the jobs available are filled by those with little or no education or training. In short, our society has moved from one based upon industrial production to one based upon educated manpower—a "knowledge society."

Dr. Thomas F. Green, Director of the USOE-supported Educational Policy Research Center at Syracuse University says that the chief characteristic of the knowledge society is that it requires knowledge-based skills rather than craft-based skills.[2] He predicts that the post-industrial society is likely to require an enormous expansion in learning, but not necessarily education. What is required is not college degrees, but skills; and Green sees an increase in non-degree programs, an expansion in education for adults at all stages of their lives, and a multiplication of proprietary schools that educate specific skills. Man's marketable skills, says Green, will no longer be tied to a specific set of tasks within an organization. Knowledge skills, the result of diverse training and practice, are polyvalent and applicable to a wide range of tasks. Work will need to be organized for its education value instead of education for its work value. An individual will have multiple careers, and his education will be distributed not over longer, consecutive periods in life but over shorter spans of time throughout his entire life[3]; it is estimated that 70% of the

Office of Computing Activities, National Sciences Foundation, Washington, D.C. This paper was written while the author was with the U.S. Office of Education (USOE). However, the views expressed herein are those of the author and do not necessarily represent those of USOE or the National Science Foundation (NSF).

children entering school today will work in occupations that do not now exist.

This new emphasis on knowledge, in turn, places greater demands on education. More and more, education is a prerequisite to participation in the "knowledge society." Colleges and universities, in turn, have a responsibility to design curricula to meet the needs of the student and to educate all who enter the educational system. Mass teaching through lectures aimed at the mean of the group is yielding to a modular curriculum tailored to meet the needs of the individual. The individualization of learning and the requirement for dynamically changing curricula will require changes in the administration and management of education.

New tools are required to handle the ever-increasing body of knowledge and to provide the means for educating all students, and the computer is destined to play a significant role. It is estimated that by the end of the decade the computer and computer-related activities will account for one third of our gross national product. In addition to its influence on society, the computer will have an equally significant impact upon education.

THE GROWING DIMENSIONS OF HIGHER EDUCATION

Today the social demand for higher education is rising. Green observes[4] that probably the most startling achievement of the American education system is the consistent enlargement in the proportion of each generation attaining grade twelve. In 1910, eight of every 100 completed grade 12; in 1969, 80 of every 100 graduated and in 1976, 90% of those reaching age eighteen will have completed 12 years of schooling. How has higher education responded to this new social demand? Green reports that since 1920, except for disruptions of war and depression, (1) the ratio of bachelor level degrees to high school diplomas five years earlier has been markedly stable—slightly less than 0.3, and (2) the ratio of completion to starts has been markedly constant—approximately 0.55. He concludes that these results imply that academic standards are not based upon some absolute standard of performance but are merely used to maintain a constant ratio between starts and completions.

While in the past, many students were barred due to a lack of resources, new assistance programs and federal loans are breaking down the economic barriers to higher education. Further, the rapid expansion of junior and community colleges and other institutions with open admissions has greatly expanded the opportunities now available in higher education. In the school year 1955-56 there were 2.6 million students enrolled in credit courses in our colleges and universities; in 1967 that figure rose to 6 million, and is projected to be 9 million by 1975. In 1967, $16.8 billion was invested in higher education, 23% obtained from federal funds; these figures are expected to rise to $34 billion to $40 billion and 30-50% by 1975.

These numbers reflect an unprecedented growth rate within our colleges and universities as well as an increase in the number of such institutions. The demand upon the limited academic resources (human and physical) has been great. The need for new approaches to provide a quality education to the ever increasing number of students is one of the nation's most pressing problems.

THE GROWING CONCERN FOR THE QUALITY OF EDUCATION

Today we live in a "credentials society" where education is often equated with academic degrees and the number of years of schooling. In the knowledge society performance rather than credentials will be the criterion of learning. On campuses throughout the country more and more attention is being paid to an evaluation of the quality of education.

Using a sample of students from 248 accredited four-year colleges and universities, Astin[5] analyzed the assumption that intellectual development is enhanced by attendance at a "high-quality institution" by comparing scores of students who had taken the National Merit Scholarship Qualifying Tests while in high school and their subsequent scores on the graduate record examination. He found that student achievement was not facilitated either by the intellectual level of his classmates or by the level of academic competitiveness or financial resources of his instituion. The major finding is that the differences in student achievement during the senior year were much more highly dependent upon variations in student characteristics that existed before entrance into college than upon characteristics of the undergraduate college attended. Astin suggests that the assumption of institutional excellence as it relates to the intellectual development of the student be reexamined; that the concept of "value-added" would be a better criterion for evaluating colleges and universities in the performance of their educational mission.

There are a growing number who feel that evaluation of student performance should not be made on the basis of the normal curve and normative comparisons but instead should be based upon skills and performance necessary to reach specifiable objectives. At the same time, the information explosion has been both a blessing and a curse to education. It has been estimated as much technical knowledge will be developed in the next 30 years as has been accumulated in the entire history of mankind. The problems of developing curricula have greatly multiplied as old disciplinary lines vanish and new disciplines are created. The rapid increase in knowledge has significantly reduced the half-life of information.[6]

Rosenstein and Cromwell point out that curricula must be designed to assist the student to face problems that will occur sometime in the future.[7] Existing curricula seldom keep pace with current knowledge. In an analysis of the University of California at Los Angeles engineering curriculum they found little correspondence between the types of engineers being graduated and regional and national employment needs. They did find a high amount of redundancy within the curriculum and noted that Hooke's Law had been taught seven times, each time as though it had not been taught before. Rosenstein and Cromwell feel that a dynamic curriculum must be provided with feedback on a regular two-year basis and recommend (1) that surveys be conducted to furnish the national career profile of practicing professionals and to allow local correlations between career profiles of an institution's graduates and the effectiveness of past curricula, (2) that 3-4% of faculty time be devoted to research upon the quality control of the educational product, and (3) that computer-accessed files be maintained for storage and retrieval of all curriculum information.

The rapid change in information has lead to more modular curricula, packaged around concepts, thereby making it relatively easy to add new concepts to an expanding curriculum and remove dated concepts. The creation of a curriculum around concepts make it more convenient for interdisciplinary study. Modular concept packages will make it possible for a student to build his own individual curriculum. It is conceivable that colleges and universities will offer several thousand concept modules with only a small number being required skills and will permit the student to select and develop his curriculum along the lines of his interest.

Education is a continuing process in the knowledge society, and the modular form lends itself to the mini-course and the weekend or vacation-time study institute for professionals not attending the university. The freedom of choice and learner control of the materials could be the educational mode of the future. Computers with graphic displays capable of branching and providing immediate feedback; variable-speed audio tapes that permit learners to speed-up speech to twice the normal rate without major losses in comprehension; and multiple-screen video presentations that can be coordinated to provide large quantities of visual information stimultaneously are all technological advances in information transfer that permit the human learner to obtain and absorb vast quantities of information more accurately and at far more rapid rates than the more traditional linear methods such as the lecture.

Educational technology can provide the learner with more information, more accurately, more effectively than the teacher. In the past, prior to the information explosion, teachers pieced together information that was not widely accessible and presented it to students through a lecture. Today modern information technology reduces the need for the professor to deliver timely information to his class and requires a shift in his teaching style. He now discusses the more theoretical and heuristic-type information in a seminar, and the student obtains information by means of educational technology.[8]

With increasing quantities of information having short life-spans, the costs of curricula are increasing. One approach to the increasing costs is the use of educational technology. Cost-effectiveness studies indicate that educational technology approaches economic efficiency when designed to cover large regions and when used by many students. Large centers, such as the Educational Resources Information Centers, Medical Information Libraries and Chemical Abstracting Services, are being established, whose function it is to search out all forms of educational information and instructional materials and to make them rapidly available on demand to remote educational sites through modern telecommunications.

In this way, there is no need to duplicate expensive libraries and facilities in each geographic region; there is no need for each college to develop a comprehensive curriculum duplicating the activities of countless other colleges; there is no need to be physically on campus nor any need to access the information durin_ ι specific hour in the day. The centralization of resources also permits the concentration of funds necessary to produce high-quality educational materials. In sum, the centralization of resources reduces duplication and the fractionalization of educational efforts and permits the pooling of economic resources necessary to develop high-speed distribution networks and of high-quality, high-cost materials for all students within a large geographic area on an individual demand basis.

FROM MASS TEACHING TO INDIVIDUALIZED LEARNING

In the past teachers taught and students passed or failed. Today, students come to learn, and it is educational systems that pass or fail. There is a strong trend toward designing curricula in a manner to meet the needs of the individual student while adapting it to meet the necessity of group instruction. Individually Prescribed Instruction (IPI) is a method in which learning materials are designed to meet specified behavioral objectives.[9] The student usually progresses at his own rate. In addition to the IPI project begun at the Oakleaf School in Pittsburgh by the USOE-supported Learning Research and Development Center there is the Learning Activities Packages (LAP) developed at t' Nova School in Fort Lauderdale, Florida; and Project PLAN (Program for Learning in Accordance with Needs) developed by the American Institutes for Research and Westinghouse Learning Corporation.[10]

F.S. Kelly,[11] C.B. Ferster and M.C. Perrott have combined the principles of operant reinforcement with individual progress into a college-level course.[12] Rejecting the notion that learning should be broken down into small packages which must be completed before proceeding to the next one, they believe that learning is a resultant of many responses to larger and more general concepts. The procedure used in their approach is for the student to master a segment of a prepared text and then schedule an interview with another more advanced student who listens without interruption while noting omssions and inaccuracies. After a discussion, if both students are satisfied that the learner has mastered the concept, the results are recorded on a progress chart and the learner proceeds to study next concept. If either is dissatisfied, the learner is rescheduled for another session.

The essential characteristics of individualized instruction are (1) definition of goals or objectives, (2) immediate feedback to the student, (3) mastery of a concept before proceeding to the next concept, (4) progression at the student's own pace, (5) providing an individual curriculum to meet the student's interest and needs, and (6) providing the instructor with a history of student performance. To supply this broad spectrum of characteristics we must necessarily have recourse to technological aids to education; in particular, to the computer.

THE COMPUTER AND FEDERAL SUPPORT

The computer today is a basic industry affecting every aspect of society. The Department of Commerce estimates that for 1968 the United States computer shipments totaled $5.5 billion with another $5.95 billion for software. In 1969 computer shipments increased 17% to $6.4 billion,[13] colleges and universities representing about 5% of the market. It is estimated by Dr. H.R.J. Grosch, Director of The Computer Science and Technology Center of the National Bureau of Standards that by the end of the decade computers will be involved one way or another in 30% of our gross national product.[14] The Organization for Economic Cooperation and Development estimates that the figure may be as high as 50% of the gross national product.

The federal investment in the support of computers in education has been a minor one. The National Science Foundation (NSF) estimates that since the fiscal year 1965 federal support of computers in education has increased by about $20 million. However, during the same four-year period, the federal government's share of the percentage of total funds provided has decreased from 36% to 23% and non-federal support has increased from 34% to 54%. In other words, institutional support for computers is much larger than federal support. The higher education community has been the prime mover in expanding the role of the computer in education.[15] Almost all major universities and more than a third of four-year colleges provide computing services for research and instruction. It is estimated that 60% to 70% of all college students are enrolled in institutions at which there is a computer of some kind being used for instruction.

In February 1967 a panel headed by Dr. John R. Pierce on Computers in Higher Education made a report to the President's Science Advisory Committee and recommended that colleges and universities in cooperation with the federal government take steps to provide computing services for higher education.[16] In July 1967 NSF established the Office of Computing Activities (OCA) as a response to the Presidential Directive to work with the USOE in an experimental program to develop the potential of computers in education.

In fiscal year 1968 OCA spend $22 million and $17 million in fiscal year 1969. Within the OCA are three major programs:

Computer Science and Engineering–basic research on theoretical computer science, software and programming systems and systems design.

Computer Innovation in Education–regional cooperative computing, computer-oriented curricular activities, and applied systems research.

Computer Applications in Research–emphasis is placed on advanced computational techniques and systems for specific research applications.

It is estimated that the USOE has spent approximately $67 million on research and instructional uses of computers in education.[17] While the USOE is authorized to support the development of computer networks, the purchase of computers, the training of computer personnel and research and development of computers in education, the reduction in funds available has greatly curtailed its participation in computers in higher education.

The Networks for Knowledge, Title VIII of the Higher Education Act, provides for the establishment and joint operation of electronic computer networks to participating institutions for such purposes as financial and student records, student course work, or transmission of library materials. Unfortunately, no money has been provided for this activity. Title VI of the Higher Education Act allows for the purchase of computers on a 50-50 cost-sharing basis. In fiscal year 1970 no funds were available for equipment. The Education Professions Development Act of 1967 allows for the training of educational personnel and can be used to support computer training; six training institutes were provided in 1970. The Cooperative Research program provides for research and development in education. The support of computer-related research under these provisions reached its peak in fiscal year 1967 when 72 projects totaling $5.6 million were supported. Since then there has been a steady decline–50 projects for $4.9 million in fiscal year 1968; 27 projects for $3.6 million in 1969 and an estimated 6 projects for $1.1 million for fiscal year 1970.

COMPUTER PROJECT SUPPORT

While the range of projects supported by NSF and USOE is large, a few examples may convey the scope of the activities being undertaken.

Management Information Systems (MIS)

The concepts of accountability and cost-effectiveness are permeating the thinking of most colleges and universities. Many schools are changing from incremental budgeting to input-output models of accounting. Essential to this type of activity is a management information system. In order to facilitate this trend and to develop compatible systems the USOE is supporting the Western Interstate Commission for Higher Education (WICHE) to:

> (a) Design, develop and implement management information systems and data bases including common data elements at local and state levels including community colleges, colleges, universities and higher education coordinating agencies, both public and private, which will improve the capability of these institutions and agencies to allocate resources more effectively and provide them, on a continuing basis, with comparable data on the costs of instructional programs by level of student, level of course, and field of study;
>
> (b) Begin the task of identifying institutional input-output indicators for instruction, research and external service programs, indicators of both quality and quantity and relating varying educational costs to such indicators;
>
> (c) Conduct an educational program including instruction and seminars in systems analysis, operations research, program budgeting, cost-benefit analysis, and the use of simulation models in training administrators in the use of these tools in decision-making.

The primary task will be to establish a standard compatible set of data elements which will be of the greatest practical universality and flexibility so that all levels of institutions and any individual institution can use on a common and consistent basis those parts of interest beyond the requirements of participation. Similarly, allowance must be made for suitable aggregation of the data so that they may be used for review purposes at echelons above the campus level.

Networks

Several major efforts are being supported in an attempt to evaluate the potential of computer networks to provide computer capacity throughout higher education. In 1968-69 the NSF established fifteen regional computer network experiments.[18] These centers* were created to serve as models and to attempt to assess the role and costs of computing activities. They include 12 major universities, 116 participating colleges, 11 junior colleges and 27 secondary schools located in 21 states. They provide computing capacity, work with users to develop computer-oriented curricula, and train faculty in their own and member institutions in the uses of computers in education. In the fiscal year 1970 three new regional centers were established.

Another approach to the sharing of computer capacity is the USOE-NSF-supported Interuniversity Communications Council (EDUCOM) Educational Information Network (EIN), through which computational capabilities existing in colleges and universities will be made accessible to users in other colleges and universities. The system is based on the concept that programs operate most effectively when they are run at the installation where they were created. The network thus supports the interchange of computational services on a fee basis and in this sense complements rather than overlaps the purpose of most user groups which only interchange programs.

The initial resource of the network is a catalog containing descriptions of programs submitted by EIN members. Programs in the EIN catalog may be run at the originating installation at rates established by that installation. These rates cover processing and handling costs, to which are added the overhead costs of EIN. Ultimately, it is expected that many programs may be remotely accessed by the user at his own installation, although actually run at the originating installation.

Instructional Uses of Computers

USOE has initiated several studies on the feasibility of a computer utility.[19] The objective was to determine what computer services could be installed in a centralized facility which would service 100,000 students within a 100-mile radius at a minimal cost and to evaluate the worth of these services. Interestingly enough, independent studies showed that it would be possible to provide the administrative computer services for the schools, junior colleges and

*The regional centers are: Carnegie-Mellon University, Cornell University, Dartmouth College, Illinois Institute of Technology, University of Iowa, Oregon State University, Purdue University, St. Anselm's College, Southern Regional Education Board, Stanford University, California Institute of Technology, University of Texas, Texas A&M, State University of New York at Buffalo and the Mid-Atlantic Educational Research Center (MERC).

universities within this area during the after-school hours while providing terminals for students for problem solving, computer concept and vocational training during the school day—all of this for approximately 1% of the school's operating budget. In order to implement such a system, extensive curriculum development is required. The U.S. Office of Education's Northwest Regional Educational Laboratory is undertaking the development of prototype materials which will be field tested, revised and made available to education.

In analyzing the results of the Florida State CAI physics curriculum,[20] it was found[21] that students who finished the CAI course performed as well as students instructed by lecture. Also, persons who are less mature and not scientifically oriented in their academic style had a greater probability of success in the CAI mode of instruction, while those who were more autonomous and scientifically oriented in their method of inquiry had a greater chance of success in the traditional lecture mode of physics instruction.

THE ROLE OF THE COMPUTER IN HIGHER EDUCATION

The transition to a knowledge society is creating new demands on individuals and institutions. Higher education is moving from providing mass education for the intellectual few to providing education on an individual basis for a large heterogeneous population in a wide variety of skills and knowledge on a continuing basis with a rapidly changing curriculum. In order to meet this challenge, colleges and universities are reorganizing their management structures and seeking new ways to share limited resources. Instructional practices are being modified in order to cope with the increasing numbers of students and the ever-expanding body of knowledge. Many institutions, in addition to developing their own computer facilites, are sharing their resources and know-how with small colleges through computer networks; for as the Pierce Report concluded, the computer is as essential to a university as a library, and any student who graduates without exposure to the computer will be deficient in his education.

The computer in education is more than an automated bookkeeping system, a giant slide rule or a medium for classroom instruction, its most significant potential is as a tool to assist man to function at higher cognitive levels of understanding. The computer permits man to break the cognitive barriers that limit human processes, enlarges man's span of control over vast quantities of data and allows for time expansion or compression of observational processes. The use of graphics provides more sophisticated models for the description and manipulation of complex ideas. The use of computer information storage and retrieval networks makes information readily available to everyone.

However, if computing is to be made available to all, and if it is to become an integral part of higher education, several major problems must be overcome. Needless and costly duplication must be avoided through the sharing of resources; programs must be constructed with a view toward their exportability to other locations; new lines of communication must be created to speed new discoveries in one field to all fields.

Historically, education has been slow to change. However, today the new demands of science, society and students are converging and effecting the major changes necessary for higher education to adapt to the needs of the knowledge society.

REFERENCES

1. R. W. Tyler, "The Changing Structure of American Institutions of Higher Education," in *The Economics and Financing of Higher Education in the United States,* U.S. Government Printing Office, Washington, D. C., 1969, pp. 305-320.
2. T.F. Green, "Education and Schooling in Post-Industrial America: Some Direction for Policy", U.S. Government Printing Office, Washington, D.C., 1970.
3. T.F. Green, "Schools and Communities: A Look Forward," *Harvard Educ. Rev.*, 39, 221 (Spring, 1969).
4. T. F. Green, *op. cit.* "Education and Schooling in Post-Industrial America."
5. A.W. Astin, "Undergraduate Achievement and Institutional Excellence," *Science, 161,* 661 (August 16, 1968).
6. A.R. Molnar, "The Computer and Curriculum Analysis," delivered at the Organization for Economic Cooperation and Development, Center for Educational Research and Innovation Conference on "The Instructional Uses of Computers in Education," Paris, France, March 19, 1970.
7. A. B. Rosenstein and L. Cromwell. *Dynamic Analysis and Design of Engineering Curricula–The Information Base,* Department of Engineering, University of California, Los Angeles, California, June 1968.
8. A. R. Molnar, "The Frontiers of Computers in Society," *ANALOG/HYBRID Computer Education Users Group Transactions,* September 1969.
9. *Individualized Instruction: An Annotated Bibliography.* ERIC Clearinghouse on Educational Media and Technology Institute for Communication Research, Stanford University, Stanford, California, October 1968.
10. J. C. Flanagan, *Individualizing Education,* American Institutes for Research, Palo Alto, California, 1968.
11. F. S. Keller, "Good-bye, Teacher. . . ," *J. Appl. Behavior Anal.,* 1968.
12. C.B. Ferster and M.C. Perrott, *Behavior Principles,* Appleton-Century-Crofts, New York, 1968.
13. *1970 National Science Foundation Authorization Hearings Before the Subcommittee on Science, Research and Development of the Committee on Science and Astronautics, U.S. House of Representatives,* Vol. II, U.S. Government Printing Office, Washington, D.C., April 1969, p. 587.
14. "Research and Development," *Government Executive,* December 1969, p. 18.
15. *1970 National Science Foundation Authorization Hearings* Vol. 1, U.S. Government Printing Office, Washington, D.C., April 1969, p. 653.
16. *Computers in Higher Education, Report of the President's Science Advisory Committee,* U.S. Government Printing Office, Washington, D.C., 1967.

17. A. R. Molnar and B. Sherman, *U.S. Office of Education Support of Computer Activities,* U.S. Government Printing Office, Washington, D.C., 1969.
18. *A First Report on an Exploratory Program of Regional Cooperative Computing Activities,* Oregon State University, Corvallis, Oregon, January 1970.
19. *Feasibility Study for a Remote Terminal Central Computing Facility Serving School and College Institutions,* IBM, Washington, D.C., 1967.
A Feasibility Study of a Central Computer Facility for an Educational System, General Learning Corporation, Washington, D. C., 1968.
Functional Analysis and Preliminary Specifications for a Single Integrated Central Computer System for Secondary Schools and Junior Colleges, Computation Planning, Inc., Bethesda, Maryland, 1968.
20. O. M. Kromhout, D. N. Hansen, and G. Schwarz, "A Computer-Assisted and -Managed Course in Physical Sciences," in this volume.
21. D.N. Hansen *et al.* "Research and Implementation of Collegiate Instruction of Physics via Computer-Assisted Instruction," Florida State University, Tallahassee, Florida, November 15, 1968.

APPENDICES

Appendix A. Inexpensive Computers

An inexpensive computer is either a small computer or a desk-top computer, also known as a programmable calculator. The two Tables in this appendix contain representative information about both types of computer, although neither is intended to be comprehensive. As of this writing (February 1970) all of this information is at least one year old, hence is only intended to be indicative, especially in regard to prices.

All the computers listed in Table 1 are considered "small" by virtue of their memory sizes, with basic core storage of 8000 words or less, although all are more or less expandable. All are priced under $25,000. Although memory size is the prime indication of computing power, capacity also depends on the number of bits in each word, which determines the complexity of the coding as well as the arithmetic accuracy of the computer. The computers listed here are either general-purpose computers or particularly adapted to scientific or educational use.

Operational characteristics of most interest to users are hardware priority interrupts, direct memory access, and hardware instructions. Hardware priority interrupts stop the normal data-scanning sequence and direct attention to the priority signals. Direct memory access indicates the number of channels by which data can be entered into the memory directly, without passing through the central processor. Hardware instructions are those executed with lower demands on the processor, hence are less subject to loss than software instructions.

Software is often priced separately from the computer and is available from a number of sources. The software listed typifies programs available from the vendor and is not meant to be exhaustive. Many users design their own software or contract with consultants or software firms. Input/output devices usually depend on intended applications, and peripheral equipment is generally supplied separately from the computer.

Desk-top computers may be expected to find ever-increasing classroom use over the next decade by virtue of their economy, ease of use, and portability. Two of the papers in this volume reported specifically on such uses (Goodman, Stoner). Their cost per student-hour is extremely low when compared with any presently available computer-aided instruction system. Table 2 lists some two dozen such computers.

TABLE 1 Small Computers[a]

Mfr. & model	Memory, basic/ expandable (in K-words)	Word length (bits)	Cycle time (μsec)	Hardware priority interrupts, basic/max	Data channels	Direct memory access	Hardware instructions	Software	I/O devices furnished or available	Basic cost ($)
Business Information Technology, 5 Strathmore Rd., Natick, Mass. 01760										
480	1/65	variable	3	1/1	4	3	–	3 assemblers; complete utility systems; ASA basic FORTRAN; math pack; calc program	ASR-33	7,100
483	1/65	variable	3	4/32	4	2	125		ASR-33	6,900
Compiler Systems, P.O. Box 366, Ridgefield, Conn. 06877										
CSI-16	4/32	16	1	16/256	to 255	to 255	32	FORTRAN IV: ALGOL: basic assembler	floating-point hardware; teletype; cassette tape; disk; drum; line printer	10,750
CSI-24	4/8,000	24	1	16/256	to 255	to 255	32			14,950
Data General, Route 9, Southboro, Mass. 01772										
NOVA	4/32	16	2.6	16 levels/ 62 devices	1	1	180	assembler; editor; debug; floating-point pkg.; utility pkg.; diagnostics	ASR-33; tape reader and punch; line printer; card reader; disk; plotter	7,950

Digital Equipment, 146 Main Street, Maynard, Mass. 01754										
PDP-8	4/32	12	1.6	0/-	1	1	8 basic 20 micro	FORTRAN compiler; FOCAL; BASIC; ALGOL; floating point; signal averaging; pulse-height analyzer	ASR-33 FLPA[b]	
PDP-15	4/128	18	0.8	8 levels/ 36 levels	8	Part of I/O bus	13 basic +3 groups micro	FORTRAN IV; FOCAL; background-foreground monitor; any standard program and extensive libraries	ASR-33 teletype; FLPA	16,500

Mfr. & model	Memory, basic/ expandable (in K-words)	Word length (bits)	Cycle time (μsec)	Hardware priority interrupts, basic/max	Data channels	Direct memory access	Hardware instructions	Software	I/O devices furnished or available	Basic cost ($)
Electronic Associates, 187 Monmouth Park Hwy., West Long Branch, N.J. 07764										
EAI640	8/32	16	1.65	64/64	4	1	64	assembler; FORTRAN IV; operations interpreter; monitors; math; libraries; utility package; application software	teletypes; paper tape reader & punch; card reader; line printer disk; mag tape	--
Fabri-Tek, 5901 S. County Rd. 18, Minneapolis, Minn. 55436										
COM-TRAN-8	8/64	8	–	4/-	256	1	77	conversational FORTRAN; SIMPLE; assembler	teletype; classroom display	24,500
BI-TRAN-6	0.128/-	6	–	0/-	1	1	30	machine language		6,300
Hewlett-Packard, 1501 Page Mill Rd., Palo Alto, Calif. 94304										
HP-2114A	4/8	16	2	8/56	8	–	70	FORTRAN; assembler; modular I/O drivers; utility programs	FLPA	9,950

HP-2116B	8/32	16	1.6	16/48	16	2 opt.	70,+10 opt.	FORTRAN; ALGOL; BASIC; extended assembler; modular I/O drivers; utility programs	FLPA	24,000
Honeywell, Computer Control Div., Old Connecticut Path, Framingham, Mass. 01701										
DDP-516	4/32	16	0.96	2/50	24	4 or 20	72	over 500 programs incl FORTRAN IV	ASR-33; FLPA	25,000
H-316	4/16	16	1.6	2/50	24	4 or 20	72	over 500 programs incl FORTRAN IV	FLPA	9,700
Interdata, 2 Crescent Pl., Oceanport, N.J. 07757										
3	4/32	16	1	4/12	255	8	10 micro	symbolic assembler; interactive	teletype	12,700
4	4/32	16	1	4/12	255	8	10 micro	FORTRAN; hexadecimal debug: editor (TIDE)	teletype	15,700

Mfr. & model	Memory, basic/ expandable (in K-words)	Word length (bits)	Cycle time (μsec)	Hardware priority interrupts, basic/max	Data channels	Direct memory access	Hardware instructions	Software	I/O devices furnished or available	Basic cost ($)
Scientific Control, Box 96, Carrollton, Texas 75006										
4700	4/65	16	0.92	6/262	5	8	90-109	SPL assembler; FORTRAN compiler; real-time monitor	–	16,200
Varian Data Machines, 2722 Michelson Dr., Irvine, Calif. 92664										
520/i	4/32	8	1.5	11 lines 4 levels /-	party line I/O	DMA port available	50	symbolic assembler; debug; diagnostic & test routines; program & subroutine library	FLPA	7,500

620/i	4/32	16/18	1.8	0/64	party line I/O	10	107	symbolic assembler; ASA FORTRAN; diagnostic & utility routines; diagnostic pkg; program & subroutine library	FLPA	12,100

[a]This table was excerpted with the kind permission of the publisher from "Survey of Small Computers," in *Instruments and Control Systems*, 69, August 1969. An expanded version of this article is to appear in the March 1971 issue.
[b]Full line of peripherals available.

TABLE 2

Manufacturer and model name	Price Delivery	Weight	Number of digits input	Number of digits output	Type of output hard copy = hc	Automatic key capabilities: decimal = dc; round off = ro square root = sr; log key = lg
Busicom USA 31 E. 28th St., N.Y., N.Y. 10016 (212) 689-4925 Model 207	$2,985 stock	33	20	20	crt	dc (programmable), sr
Canon 64-10 Queens Blvd., Woodside, N.Y. 11377 (212) 478-5800 Model 162P	$1,295 stock	14	16	16	Nixie	ro
Canon 64-10 Queens Blvd., Woodside, N.Y. 11377 (212) 478-5630 Model 164P	$1,650 60 days	15	16	16	Nixie	ro, sr
Cintra 440 Logue Ave., Mountain View, Calif. 94040 (415) 968-9230 Scientist 909	$3,780 30 days	24	12	10	Nixie	ro, sr, lg
Cintra 440 Logue Ave., Mountain View, Calif. 94040 (415) 968-9230 Scientist 909-03	$4,820 90 days	24	12	10	Nixie	ro, sr, lg
Commodore 390 Reed St., Santa Clara, Calif. 95050 (408) 274-0756 Model AL-1000	$1,495 stock	23	14	14	Nixie	ro, sr
Eugene Dietzgen Co. 2425 N. Sheffield Ave., Chicago, Ill. 60614 (312) 549-3300 Model 7410PA	$3,480 stock	12	14	10	Nixie	dc (programmable), sr, lg
Friden 2350 Washington Ave., San Leandro, Calif. 94577 (415) 357-6800 Model 1152	$1,495 30 days	42	13	13	Printed tape hc	ro, sr
Hewlett-Packard Box 301, Loveland, Colo. 80537 (303) 667-5000 Model 9100A	$4,400 stock	40	12	12	crt hc (optional)	dc, ro, sr, lg
Hewlett-Packard Box 301, Loveland, Colo. 80537 (303) 667-5000 Model 9100B	$4,900 30 days	40	12	12	crt hc (optional)	dc, ro, sr, lg
IME, Inc. One IME Plaza, N. Bergen, N.J. 07047 (201) 861-3800 Model 88SR, DG-308	$3,140 stock	28	16	16	Nixie hc (optional)	dc, ro, sr
Mathatronics 241 Crescent St., Waltham, Mass. 02154 (617) 893-1630 Mathatron III	$11,420 stock	15 (keyboard)	9	9	Printed tape or page hc	dc, ro, sr, lg
Monroe International 550 Central Ave., Orange, N.J. 07051 (201) 673-3400 Model 1265	$2,495 30 days	24	14	14	Printer hc	dc (programmable), ro, sr
Monroe International 550 Central Ave., Orange, N.J. 07051 (201) 673-6600 Model 1665	$3,250 30 days	13	13	10	Nixie	dc (programmable), sr, lg
Olivetti-Underwood* One Park Ave., N.Y., N.Y. 10016 (212) 679-3400 Model Programme 101	$3,850 stock	65	22	22	Printed tape hc	dc, ro, sr
SCM, Inc. 299 Park Ave., N.Y., N.Y. 10017 (212) 752-2700 Model 588PR	$1,795 stock	55	16	16	Printed tape hc	ro, sr
SCM, Inc. 299 Park Ave., N.Y., N.Y. 10017 (212) 7[illegible]-2700 Model 1016PR	$2,495 stock	36	14	14	Printed tape hc	dc (programmable), ro, sr
Sharp Electronics 178 Commerce St., Carlstadt, N.J. 07072 (201) 933-4200 Model 361P	$1,395 - stock	17	16	16	Nixie	ro, sr
Toshiba America 447 Madison Ave., N.Y., N.Y. 10022 (212) 682-4163 Model 1623G	$1,395 30 days	12	16	16	Nixie	ro
Wang Labs. 836 N. St., Tewksbury, Mass. 01876 (617) 851-7311 Model 250 (with 4 keyboards)	$6,640 20 days	6 (keyboard)	14	10	Nixie	ro, sr
Wang Labs. 836 N. St., Tewksbury, Mass. 01876 (617) 851-7311 Model 370/362	$4,295 20 days	9 (keyboard)	14	10	Nixie	sr, lg
Wang Labs. 836 N. St., Tewksbury, Mass. 01876 (617) 851-7311 Model 700	$4,900 90 days	35	12	24	Dual nixie	dc, sr, lg
Wang Labs. 836 N. St., Tewksbury, Mass. 01876 (617) 851-7311 Model 720B	$8,800 6 months	36	12	24	Dual nixie	dc, sr, lg

[a]This table is reprinted with the kind permission of the publishers from "Programmable

Number of internal program steps / type of external program storage	Number of addressable registers / Fixed or floating point	Options, attachments
No internal—unlimited external with punched cards	5 Fixed	None
64 Hardwired at factory	2 Fixed	None
64 Punched cards	4 Fixed	None
85 Paper tape, marked cards	26 Fixed or floating	Programmer with 25,670 program steps
256 Paper tape; marked cards	100 Fixed or floating	Programmer with 25,800 program steps
30 None	4 Fixed	None
128 Punched cards	10 Floating	Card reader; printer model, 256 program steps
30 None	1 Fixed	None
196 Magnetic cards; marked cards	16 Fixed or floating	Printer; teletypewriter, typewriter; x-y plotter
392 Magnetic cards; marked cards	32 Fixed or floating	Printer, teletypewriter, typewriter, x-y plotter
512 in remote programmer Punched cards	7 Fixed	Typewriter; paper tape reader, additional registers
2,816 Punched paper tape	176 Floating	Alphanumeric terminal, punched card reader, acoustic terminal
128 Punched cards	7 Fixed	Card reader; 256 program steps
128 Punched cards	10 Floating	Card reader, printer model
120 Magnetic cards	10 Fixed	None
66 None	3 Fixed	Unlimited program steps with magnetic tape cassette
100 None	7 Fixed	Unlimited program steps with magnetic tape cassette
62 None	2 Fixed	None
17 None	3 Fixed	None
80 Punched cards	5 Floating	Programmable decision capability
100 Punched cards	14 Floating	Programmable decision capability
960 Magnetic tape cassette	122 Floating	None
1,984 Magnetic tape cassette	248 Floating	Plotter

Calculators Shopper's Guide," which appeared in *Computer Decisions* 25, July 1970.

Appendix B. Glossary

This glossary contains definitions or explanations of many of the more frequently used and less familiar terms employed by computer specialists, many of which occur in the papers presented in this volume. These are not always complete definitions and are often defined from an educational standpoint. For additional information the reader is referred to the following:

USA Standard Vocabulary for Information Processing, American Standards Association Institute, New York, 1966.

Automatic Data Processing Glossary, Bureau of the Budget, U.S. Government Printing Office, Washington, D.C., 1964. PREX 2.2:AU8/2.

Shorter glossaries, often with simplified definitions, are available from most computer manufacturers and some professional societies.

Access time: Time required to obtain information from storage or to put information away in storage.

Acoustic coupler: A device used to connect a user terminal to an ordinary telephone to transfer information to the computer by phone via an audible signal. (See "dataset," "modem.")

ALGOL: ALGOrithmic Language. A second-generation programming language with a more logical structure than FORTRAN. Three formally defined versions exist: ALGOL58, ALGOL60, and ALGOL68. Similar languages are MAD, JOVIAL, and NELIAC.

Algorithm: "A rule of procedure for solving a recurrent mathematical problem." (Quoted from *Webster's Third New International Dictionary.*)

Analog Computer: A device that uses voltages, forces, fluid volume, or other continuously variable physical quantities to represent numbers in calculations.

Analog-to-digital converter: A device that changes physical motion or electrical voltages to digital factors.

APL: A Programming Language. A highly interactive time-shared language with a full range of powerful numerical and string operators. Allows convenient manipulation of arrays and matrices.

ASCII: American Standard Code for Information Interchange. This is the character code established as standard by the American Standards Association and consisting of eight information bits plus one start and two stop bits per character, a total of 11 bits/character.

ASR: Automatic Send-Receive set. A combination teletypewriter, transmitter, and receiver, using either keyboard or paper tape. (See "half-duplex.")

Assembler: A computer program that produces machine-language instructions from a symbolic language program, usually on an item-for-item basis, producing the same number of instructions or constants as defined in the symbolic program.

Author language: A language for programming tutorial-mode computer usage for instruction.

Bandwidth: The range of frequencies that can be transmitted over a communications channel (telephone line, radio channel, etc.). The bandwidth determines the amount and quality of information that can be transmitted through the channel per second (bps). This is equivalent to cycles per second (cps) or Hertz (Hz), with the usual multiples, kilo- and mega-.

BASIC: Beginner's All-purpose Symbolic Instruction Code. An algebraic terminal language noted for its simplicity and convenience and widely used as a time-sharing language with a useful set of matrix operations.

Batch processing: A method of processing in which a number of similar input items or programs are accumulated and processed together. (See "multi-processing" and "time-sharing.")

Baud: Bits per second; a measure of transmission rate. A 110-baud channel carries 110 bits, or 10 characters per second.

Binary device: Any physical device having two states: on-off, yes-no, true-false. Most computers are made up of many elementary binary devices. (See "core.")

Bit: Contraction of "binary digit," a 1 or a 0; the smallest element of binary computer memory or logic. (See "byte.")

Branching: Altering the sequence of a program when some predesignated event occurs. Providing a change in instructional procedure as a result of an individual's performance.

Broadband: Of large bandwidth; at least bandwidth greater than that of ordinary unconditioned dialable telephone circuits, which is about 2000 Hz or bits/sec.

Broadband Exchange Service (BXS): A data communications exchange switching service offered by the Western Union Telegraph Company. The service provides full duplex facilities at bandwidths up to 4 kHz (data rates up to about 2400 bits/sec), with bandwidths up to 48 kHz a possibility in the near future. The user can dial the particular bandwidth he wants as well as the party he wishes to call, and can transmit and receive voice communications as well as data.

Buffer: A storage device used to compensate for a difference in rate of flow of data or time of occurrence of events when transmitting data from one device to another. The storage unit is often a magnetic core memory, and is referred to as buffer memory or buffer storage.

Byte: A group of bits (usually six to eight) forming a character!

CAI: Computer-Assisted Instruction; computer-aided instruction; computer-augmented instruction. Narrowly defined, it refers to tutorial or programmed instruction; defined broadly, it encompasses all instructional computer usage.

CAL: Computer-Assisted Learning.

CBI: Computer-Based Instruction.

Channel: A path for electrical signal transmission between two or more points. Also called a circuit, facility, line, link, or path.

Character: A digit, letter, or other symbol, usually requiring six to eight bits of storage. In data transmission, start and stop bits may be required for each character in addition to information bits. In the ASCII code, for instance, a total of 11 bits/character is required for transmission.

CMI: Computer-Managed Instruction. Computer administration of instructional activities. Includes grading, assignment of individual student study programs, and clerical chores.

Code. Translation of a program into machine or assembler language.

Compiler: A computer program for the translation of instructions expressed in a user language (for example, FORTRAN) into machine language.

Computational mode: In which the computer is used to calculate solutions to problems. the student doing his own programming.

Connect-time: Total elapsed time during which the user is connected directly to the computer. (See "execution time.")

Core: The rapid-access memory of a central processing unit; usually made up of many small rings (cores) of magnetic material, which may be in either of two states of magnetic polarization.

COURSEWRITER: A language for writing the text and anticipated answers of a tutorial conversation. Developed by IBM; three official versions exist for different systems. (See "author language.")

CPU: Central Processing Unit. The central section of a computer, including control, arithmetic, and memory units.

CRT: Cathode-Ray-Tube. In common use as a display device.

Cursor: A point or line of light displayed on the CRT, the position of which is under the control of either the user or the computer. The cursor is used to indicate the point at which the next display or editing operation is to occur. (See also "light pen" and "RAND tablet.")

Cycle time: The interval between the request for, and delivery of, information from storage.

Data cell: A device that stores and retrieves blocks of information on strips of magnetic tape by automatically transporting these strips from a storage cell to a reader and back. Slower than disk storage, it is less expensive and has a larger capacity (See also "diskpack" and "tape.")

Data Phone: A trade mark of the American Telephone and Telegraph Company to identify the data sets manufactured and supplied by the Bell System for use in transmission of data over the regular telephone network. It is also a service mark of the Bell System which identifies the transmission of data over the regular telephone network (Data Phone service).

Data set: A device that converts the digital pulses from a computer or an I/O terminal into audio frequency tone pulses for transmission over telephone lines or other communication channels. It is a telephone-like device, with provision for dialing another party and for voice communication with the other end of the line. (See "modem.")

Debug: To locate and correct errors in a computer program.

Diagnosis and testing mode: In which the student is tested and he or his teacher is given some interpretation of results.

Diagnostics: Routines or messages that automatically locate and define errors in computer operation or programming.

Dialogue: Tutorial mode usage in which the student assumes greater control over the selection and sequencing of messages that make up the conversation. Program files and algorithms must anticipate and respond to student queries.

Digital computer: A device that uses digits to express numbers and special symbols in calculations and other information processing. The most common type of computer. (See "analog computer.")

Direct access: See "random access."

Disk: A storage device resembling a phonograph record and coated with magnetic material for retention of information. Bits can be stored upon it and read while revolving at high speeds.

Diskpack: A stack of disks. (See also "drum" and "data cell.")

Down-Time: Time when a machine is not available for operation because of machine failure.

Drum: A cylinder coated with magnetic material for the storage of information. Bits can be stored upon and recovered from it while it is revolving at high speeds. Data transfer is usually faster than with tapes or disks.

Duplex or full duplex: In communications pertaining to simultaneous two-way, independent transmission in both directions. In contrast to "half duplex."

EDP: Electronic Data Processing.

Editor: A program for changing instructions to the computer while they are resident on some storage device.

Execution time: Time during which the computer is actively processing the user's job. Roughly equal to "connect-time" in batch-processing, it is only a small fraction of it in multiprocessing.

Facsimile (FAX): Transmission of photographs, maps, diagrams, etc., over communication channels. The image is scanned at the transmitter, reconstructed at the receiving station, and duplicated on some form of paper.

File: Any collection of similar items of information, whether stored on tape, disk, drum, cards, or in core.

Flip-flop: An electronic device having two input lines, two ouput lines, and two stable states such that a signal exists on either one of the output lines if the last pulse received by the device is on the corresponding input line. Such a device can store one binary digit. More generally, any bistable device storing one binary digit.

Flowchart, or flow diagram: A graphical representation of the sequence of operations in a computer program by means of a block diagram

Foreign exchange service: A telephone exchange service in which the subscriber's telephone (or data station) is effectively located in a distant exchange area. An unlimited number of calls or data connections can then be made within the distant exchange area for a flat monthly charge. This service is limited to the usual telephone line data rates of less than 2000 bits/sec.

FORTRAN: FORmula TRANslation. A programming language designed by IBM in 1957 for scientific use, it is now widely adapted for other uses as well and available on almost every computer in at least one of its several dialects.

Frame: The smallest unit of programmed instruction. Also, the time needed to transmit either bits or bytes of data along with other control information.

Function codes: Codes that operate machine functions, such as carriage returns, shifts etc.

Half-duplex: Pertaining to an alternate, one-way-at-a-time transmission facility (sometimes referred to as a "single"). In contrast to "duplex" or "full duplex."

Hardware: Computer components and equipment.

Heuristic: An approach to a problem solution by trial and error, instead of by a direct algorithm. A useful problem-solving strategy for learning.

High-speed: Input/output devices whose read/write speed is compatible with the speed of data processing by the computer and can be used on-line.

Hybrid computer: A combined digital and analog computer.

Information retrieval: A branch of computer science which deals with storing and searching large quantities of information.

Input device: One that brings data to be processed into the computer, such as a card reader, tape reader, keyboard, etc.

Interactive mode: Computer use in which particular attention is given to the interaction between the user and program (or system). The user's responses select various branches in the program while it remains active in the computer. For effective use response times of ten seconds or less are necessary.

Interface: A shared boundary; for example, the boundary between two subsystems or two devices. An interface unit in a remote terminal computer system is a hardware component which "matches" devices to each other; for instance, a computer to several telephone lines from remote terminals or a remote station to a telephone line. The unit may contain buffer memory, switching logic, and/or multiplexing facilities.

Interpreter: A program for user-language translation (e.g., BASIC to machine language). Roughly equivalent to a "compiler," except that each statement is executed immediately following translation, providing rapid response to small steps and a dynamic working environment. (See "translator.")

Interrupt: A hardware feature that allows the computer to stop working momentarily on one task, handle the interrupting task, and return to the first without losing information or interim results of processing.

INWATS: Similar to WATS (Wide Area Telephone Service–see below), but allows inward calls at a flat monthly rate.

I/O: Input/output of information to and from computers. Usually refers to devices such as electric typewriter, card reader and punch, paper tape reader and punch, etc.

JOSS: An algebraic programming language developed by the RAND Corporation for use with time-shared terminals. It is designed to provide rapid feedback of grammatical errors made by the programmer. Variations exist under many different names (PIL, CAL, ISIS, TELCOMP, IITRAN, and TINT).

Joy stick: An electromechanical device for moving a cursor about on a CRT, thereby indicating position to the computer. It may look like an airplane control stick in appearance. (See also "light pen," "mouse," and "RAND tablet.")

K: Denotes one thousand (kilo); when used to describe computer memory K means $1024 = 2^{10}$.

Language: A set of symbols, characters and words; their definitions and rules for usage in communicating with the computer to perform desired operations.

Latency: In a serial storage device, the time required to find the first bit in a given location. Can also refer to the elapsed time from the display of an instructional stimulus to the start (or completion) of a student's response. Frequently used as a measure of student performance in educational research.

LDX: Long Distance Xerography. A name used by the Xerox Corporation to identify its high-speed facsimile system. The system uses Xerox terminal equipment and a wide-band communication channel.

Library: A collection of programs and routines, usually tested and applicable to many varied uses. Every well-equipped computer center has one.

Light pen: A photosensitive device used for communication with a computer. When held against the face of a cathode-ray-tube, its position can be sensed. For instance, if the computer is suitably programmed, the position of the light pen can be interpreted in terms of a coordinate grid. The device can be used in this way to enter arbitrary graphical information, sketched by the operator on the CRT, into the computer memory in digital form.

Linear programming: The simplest form of programmed instruction; questions and answers, with no branching. Also a class of techniques for optimizing solutions to an incomplete set of simultaneous linear equations with respect to certain constraints on some or all of the variables.

Line switching: The technique of temporarily connecting two lines together, so that remote stations may interchange information directly.

Link: The same as a "channel."

Load: To fill the internal storage of a computer from some external storage medium.

Logic: The interconnection of bistable circuit elements within the computer.

Loop: Repeated execution of a sequence of instructions until some terminal condition is achieved. At the heart of all recursive and iterative mathematical operations as performed by the computer.

Machine language: A combination of binary digits which can be read directly by the computer without further processing and signifying basic operations such as add, compare, store, etc. (See "object" and "source" languages.)

Magnetic core: See "core."

Mark sensing: Detecting special pencil marks entered on a card, and automatically translating them into computer input.

Memory: See "storage."

MODEM: MOdulator/DEModulator. A device that converts a digital signal to an analog (frequency) signal or vice versa. Used to couple a terminal to a telephone line.

Module: A piece of equipment with specific storage capacity; an easily interchangeable item of hardware; a block of storage.

Monitor: The program to control the operation of the routines that comprise the schedule of machine runs in order to use computer time most efficiently.

Mouse: A small box moved about manually on a horizontal surface which causes a cursor to move correspondingly on a CRT screen. (Also see "joy stick," "RAND Tablet," and "light pen.")

Multiplexing: The division of a transmission facility into two or more channels, over each of which independent signals or data may be transmitted. Thus a single phone line to a remote computer may serve many terminals simultaneously.

Multiprocessing: A method of operation in which a computer facility is shared by several users. Each device actually services one user at a time, but high speed and multiple components give the appearance of handling many users simultaneously, thus also maximizing the utilization of hardware.

Nixie: A glow-discharge tube that converts a combination of electrical impulses into a visual number.

Object language: The machine language developed by the compiler from the source language.

Off-line: Processes or devices not under the control of the CPU or when used independently of the CPU.

On-line: Devices connected directly to the CPU or processes performed under its control.

Operating system: The collection of programs that direct or supervise the utilization of processing components and the execution of other programs. (See "software.")

Output device: One that translates data stored in the computer into usable information, such as card and tape punches, tape recorders, and line printers.

Overflow: Generation of a quantity which is beyond the capacity of the assigned storage.

Partitioning: Using a computing system as functionally divided into two or more subsets assigned to different tasks with different priorities.

Patchboard: A removable board containing electrical terminals to which short wires (patch cords) are connected, thereby determining the program for the machine. Generally used with analog and peripheral equipment.

PL/1: Programming Language/One. Developed by IBM to combine the features of an algebraic and a symbolic or list-processing language. Thus it can use the programs in both linguistic and computational modes.

PLANIT: Programming Language for Interaction and Teaching. Similar to COURSEWRITER, but provides more facility for on-line computation and problem solving.

Plasma panel: A flat, gas-filled panel for graphic display by computer-activated gas discharges.

Polling: A centrally controlled method of calling or interrogating a number of remote stations sequentially, to permit them to transmit information.

Port: Physical connection for a telephone line coming into the computer from a user terminal.

Processor: CPU. Also a program that will compile, assemble and execute a source program.

Program: A plan or procedure for the automatic solution of a problem by the computer. Also the list of instructions that accomplishes this.

Programmed instruction: A step-by-step presentation of instructional materials. May also include branching and concurrent testing at each step.

Programming language: Any system of instructions for presenting programs to the computer. (See "language.")

Random access: Access to storage in which the next location from which data are obtained is independent of all previous locations accessed. Storage for which access-time is approximately the same for all locations.

RAND tablet: A metal writing surface developed by the RAND Corporation for input of graphic information to a computer through the use of a special writing stylus.

Real-time: Performance of data processing during the actual time the physical process producing the data takes place, in order that results of the computation can be used to guide the physical process.

Refresh: To re-draw the image on a CRT 30-60 times per second, so that it persists without flicker. Scanning logic and buffer storage drive the CRT. Images are altered by changing information in buffer storage either directly through an analog-to-digital converter or through the CPU. (See also "storage oscilloscope," "light pen," "joystick," "mouse," and "RAND tablet.")

Remote access: Use of a computer through I/O equipment distant from the CPU, generally through the medium of a telephone line or microwave link.

Response time: The amount of time elapsed between generation of an inquiry at a data communications terminal and receipt of a response at the same terminal.

Routine: A sequence of instructions that direct the computer to perform a desired operation or set of operations.

Sequential access: Storage in which items of information become available only in sequential order, one after another, e.g., punched card decks or magnetic tapes.

Simulation: Computer usage in which the output simulates experimental results, random processes, etc.

Software, computer: Computer programs, as contrasted to hardware.

Software, instructional: Curriculum materials as contrasted with educational facilities.

Source language: A language such as FORTRAN or BASIC from which machine-language instructions are developed by compilers and assemblers.

Station: One of the input or output points on a communications system.

Storage: The capacity of a computer to save information for future use. Also the devices that save the information.

Storage oscilloscope: A device in which displays are stored on the face of the CRT by electrostatic means. In order to change any piece of the stored information, the entire display must be erased and re-formed, making this device slower in response than refreshed CRT displays.

Storage protect: A hardware feature that prohibits one user in a shared system from destroying the information stored in memory allocated to other users.

String: A connected sequence of characters, words, or other elements.

Student station: I/O equipment designed for student use, to permit the student to interact with the computer.

Teaching logic: Pattern or strategy for instruction; each author language usually implies another logic.

Teleprinter: A term referring to the equipment used in a printing telegraph system. A teletypewriter.

Teleprocessing: A form of information handling in which a data-processing system utilizes communication lines between stations and the processing unit (computer).

Teletypewriter exchange service: An automatic exchange switching service provided on a nationwide basis by the American Telephone and Telegraph Company (TWX) and the Western Union Telegraph Company (Telex).

TELPAK: A wide-band, leased line information transmission service offered by communications common carriers.

Terminal: A device attached directly to a computer through telephone lines, cables, or other communications links, and designed for user-computer interaction. Terminals may be located adjacent to the computer or at a remote location.

Tie line: A private communication channel of the type provided by communications common carriers (e.g., AT&T or Western Union) for linking two or more points.

Time-sharing: A method of operation in which a computer facility is shared by several users for different purposes at (apparently) the same time. Although the computer actually services each user in sequence, its speed makes it appear that the users are all handled simultaneously. (See "multiprocessing.")

Translator: A program that accepts statements or instructions written in one language and produces statements written in another language, or produces direct instructions to the computer for execution.

Turn-around time: The time required for completion of a job from submission to receipt of final results.

Tutorial mode: Instructional computer usage in which the machine converses with the student. More narrowly, that usage in which the author maintains the initiative by defining objectives and describing the subject matter in detail.

Underflow: Generation of an arithmetic quantity too small for the capacity of the assigned storage.

Vector generation: The capability of drawing a line between two specified points on a terminal display device.

Voice grade channel: A communication channel suitable for transmission of speech, digital or analog data, or facsimile, generally with a frequency passband of about 300 to 3000 Hz. Voice grade channels can carry data at rates greater than about 2000 bits/sec only if they are specially conditioned.

Wide Area Telephone Service (WATS): A service provided by telephone companies which, by the use of an access line, permits a customer to make an unlimited number of calls to telephones or data sets in a specific geographical zone for a flat monthly charge.

Word: A set of bits sufficient to express one computer instruction. Word lengths may be fixed or variable, depending upon the particular computer and are generally from 12 to 66 bits long. The word is generally treated as the unit of information in the transfer, processing, and storage of data.

Appendix C.
Suggestions for Further Reading

Adams, E.N., "Reflections on the Design of a CAI Operating System," *AFIPS Conference Proceedings, Spring Joint Computer Conference*, 1967.

Albrecht, R.L., E. Lindberg, and W. Mara, *Computer Methods in Mathematics*, Addison-Wesley, Reading, Mass., 1969.

American Society for Engineering Education (ASEE), Symposium on Applications of Technology to Education, 1969.

Andree, R.V., *Computer Programming and Related Mathematics*, Wiley, New York, 1967. A good book for the beginning student.

Atkinson, R.C., and H.A. Wilson, "Computer-Assisted Instruction," *Science 162*, 73, 1968.

Baker, W.O., "Computers as Information–Processing Machines in Modern Science," *Daedalus*, Fall, 1970, p. 1088. An imaginative article which indicates possibilities for the future.

Bashkow, T.R. (ed.), *Engineering Applications of Digital Computers*, Academic Press, New York, 1968.

Bernstein, J., *The Analytical Engine: Computers–Past, Present, and Future*, Random House, New York, 1964.

Blum, R. (ed.), *Computer-Based Physics: An Anthology*, Commission on College Physics, College Park, Md., 1969. A collection of nine computer-oriented teaching units for undergraduate mechanics and modern physics. Includes many worked examples and program listings.

Blum, R., and A.M. Bork, "Computers in the Physics Curriculum," *Am. J. Phys., 38, 959* (1970). A comprehensive survey.

Blum, R., " 'Image Methods in Electrostatics' (A Computer-Animated Film)," *Am. J. Phys. 36*, 412 (1968).

Blum, R. (ed.), *Introductory Computer-Based Mechanics: A One-Week Sample Course*, Commission on College Physics, College Park, Md., 1969. A complete teaching unit based on the simple harmonic oscillator by A.M. Bork, A.W. Luehrmann, and J.W. Robson.

Blum R., and R. C. Daugherty, *Short Films for Physics Teaching*, Commission on College Physics. College Park, Md., 1971. Contains a listing of computer-animated film loops.

Bork, A.M., *FORTRAN for Physics*, Addison-Wesley, Reading, Mass., 1965.

Bork, A.M., "Quantum-Mechanical Oscillator," *Am. J. Phys. 34*, 470 (1966). A computer-animated film.

Prepared by Ronald Blum, Commission on College Physics, with the assistance of Carl E. Heilman, Division of Science and Mathematics, Pennsylvania Department of Education, Harrisburg, Pennsylvania.

Bowles, E.A. (ed.), *Computers in Humanistic Research*, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

Browne, M.E. and T.E. Ivgerson, "Laser Roulette and Other Diversions," *Am. J. Phys. 38*, 723 (1970). Games to add interest, some using the computer.

Brudner, H.J., "Computer Managed Instruction," *Science 162*, 970 (1968).

Bush, V., "As We May Think," *Atlantic Monthly*, Vol. 176, No. 1, July 1945, pp. 101-108.

Bushnell, D.D., and D.W. Allen (eds.), *The Computer in American Education*, Wiley, New York, 1967.

Caffrey, J., and C.J. Mosmann, *Computers on Campus*, American Council on Education, Washington, D. C. 1967.

Campion, D., *Computers in Architectural Design*, Elsevier, London, 1968.

Commission on Education, National Academy of Engineering, *Educational Technology in Higher Education: The Promises and Limitations of ITV and CAI*, National Academy of Engineering, Washington, D.C., 1969.

Commission on Instructional Technology, *To Improve Learning: A Report to the President and the Congress of the United States*, U.S. Government Printing Office, 1970.

Computer-Assisted Instruction Guide, Entelek, Inc., Newburyport, Mass., 1968.

Computer-Based Math Laboratory, Entelek, Inc., Newburyport, Mass., 1969.

"Computer Directory and Buyer's Guide Issue," *Computers and Automation*, November 1970. A very comprehensive listing of computers, their characteristics, languages, producers and users.

"Computer Instruction Issue," *J. Chem. Educ.*, February 1970, entire issue.

"Computer Peripherals," *Control Engineering*, February 1970, p. 87. Much valuable information on equipment.

"Computers," *Eng. Educ.*, January 1971, entire issue. A diversified compendium on computers in engineering.

"Computers and Education," *Computers and Automation*, March 1970, entire issue. These articles discuss various phases of computer use in education.

Computers in Higher Education, President's Science Advisory Committee, The White House, Washington, D. C., 1967.

"Computers in Mathematics," *The Arithmetic Teacher*, March 1969, entire issue. The computer in the elementary grades.

"Computers in the '70's," *Science Journal*, October 1970, entire issue. A general collection of articles about the computer and its effects from the British point of view.

Computers in Undergraduate Education: Mathematics, Physics, Statistics, and Chemistry, Science Teaching Center, University of Maryland, College Park, Maryland, 1967.

Cuadra, C. (ed.), *Annual Review of Information Science and Technology–Volume 4*, Encyclopedia Britannica, Inc., Chicago, 1969.

"Curriculum 68, Recommendations for Academic Programs in Computer Science," *Communications of the ACM*, Vol. 11, No. 3, March 1968, pp. 151-197.

"Data Service Center Directory," *Data Systems News*, August–September, 1970, insert.

Davis, J.C., "Make Way for Minicomputers," *Chem. Engrg.*, March 8, 1971, p. 58. A discussion of computer use in the chemical process industries.

Dickson, T.R., *The Computer and Chemistry*, Freeman, San Francisco, 1968. This book has a general introduction, also useful for physics and mathematics.

Digital Computer Needs in Universities and Colleges, National Academy of Sciences–National Research Council, Washington, D.C., 1966.

"Digital Computers in Physics," *Phys. Today*, July 1970, entire issue. A useful and informative collection of articles.
"Editor's Note: Instructional Uses of the Computer," *Am. J. Phys. 35*, 273 (1967).
"Electronic Calculator Survey," *Data Systems News*, October, 1970, insert.
Elmer, R.P., "Computers... How and Why to Use Them in High School?", *The Science Teacher*, November, 1969, p. 69. Of general interest.
Feingold, S.L., "PLANIT–A Flexible Language Designed for Computer-Human Interaction," *AFIPS Conference Proceedings, Fall Joint Computer Conference*, 1967.
Fenichel, R., *et al.*, "A Program to Teach Programming," *Communications of the ACM*, March 1970, p. 141.
Feurzeig, W., "The Socratic System: A Computer System to Aid in Teaching Computer Concepts," Bolt Beranek and Newman, Cambridge, 1965.
Finerman, A. (ed.), *University Education in Computing Science*, Academic Press, New York, 1968.
Forsythe, A.I., T.A. Keenan, E.I. Organick, and W. Stenberg, *Computer Science: A First Course*, Wiley, New York, 1969. Suitable for high school as well as college courses. Extensive *Teacher's Commentary* available.
Foster, D.F., "Computer Science as a Liberal Art," *Computers and Automation*, March 1970, p. 36. This article suggests a course in modeling and simulation useful in many academic areas.
Frye, C.H., "CAI Languages: Capabilities and Applications," *Datamation*, September 1968.
Gerard, R. (ed.), *Computers and Education*, McGraw-Hill, New York, 1967.
Goldberg, A., *et al.*, "Computer-Generated Motion Pictures of One Dimensional Quantum–Mechanical Transmission and Reflection Phenomena," *Am. J. Phys. 35, 177 (1967).*
Golden, J.T., *FORTRAN IV, Programming and Computing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1965. An excellent text with numerous worked examples of interest to science students.
Goldstine, H., and J. von Neumann, *Planning and Coding of Problems for an Electronic Computing Instrument*, Institute for Advanced Study, Princeton, New Jersey, 1947.
Hamblen, J.W., *Inventory of Computers in U.S. Higher Education*, Southern Regional Education Board, Atlanta, 1969.
Hamilton, W.G., "The Revolution in Crystallography," *Science*, July 10, 1970, p. 133. The computer is helping to revolutionize the study of crystallography.
Hammersmith, A.G., "Report on Time-Sharing Services," *The Electronics Engineer*, May 1970, p. 88. A very complete listing.
Heiserman, D.L., "Computer Time-Sharing," *Electronics World*, March 1970, p. 42.
Heiserman, D.L., "Graphic Computer Terminals," *Electronics World*, October 1970, p. 34; November 1970, p. 35. An informative two-part article on a variety of video display terminals.
Holtzman, W. (ed.), *Computer-Assisted Instruction, Testing and Guidance*, Harper and Row, New York (in press).
Hughes, H.S., "Gauss, Computer-Assisted," *The Mathematics Teacher*, February 1971, p. 155. Use of the computer to study constructibility of polygons.
"IBM 1500 Operating System: COURSEWRITER II," published by IBM Corporation, New York, 1966.
Information, A Scientific American Book, W.H. Freeman, San Francisco, 1966.
"Information Processes," *Eng. Educ.*, March 1970, p. 735. Suggests useful reference materials, many of them free.

Iverson, K.E., *Elementary Functions*, Science Research Associates, Chicago, 1966.

Johnson, D.C., L.L. Hatfield, J.W. Walther, D.E. LaFrenz, P.W. Katzman, and T.E. Kieren, *Computer-Assisted Mathematics Program*, Scott, Foresman, Glenview, Ill., 1968.

Jordain, P.B. (ed.), "Condensed Computer Encyclopaedia," McGraw-Hill, New York, 1969. An informative and understandable guide to the computer.

Kimber, R., *Automation in Libraries*, Pergamon Press, Oxford, 1968.

Kemeny, J.G. and T.W. Kurtz, "Dartmouth Time-Sharing," *Science, 162*, 223 (1968).

LaFrenz, D. and T.E. Kiersen, "Computers for All Students, A New Philosophy of Computer Use," *School Science and Mathematics*, January 1969, p. 39. Refers to both mathematics and science.

Levien, R.E., *et al., The Emerging Technology, Instructional Uses of the Computer in Higher Education*, The Rand Corporation, Santa Monica, Calif., 1970. A comprehensive and detailed study sponsored by the Carnegie Commission and the National Science Foundation and addressed to "higher educational policy makers in government, in the colleges and universities, and in industry."

"Libraries and Automation," *Datamation*, February 1970, entire issue. An area of computer use that overlaps all disciplines.

Licklider, J.C.R., *Libraries of the Future*, The MIT Press, Cambridge, Massachusetts, 1965.

Lipsitz, L., "The Computer and Education," *Educational Technology*, March 1970, entire issue. Many points of view are considered in a diversified collection of articles.

Lunetta, V., and O.E. Dyrli, "Computers in the Science Curriculum," *Science Eduation*, April 1970, p. 141. These suggestions are directed chiefly to the secondary schools and to the physical sciences.

Matthews, M.V., "Choosing a Scientific Computer for Service," *Science 161*, 23 (1968).

Matthews, M.V., *The Technology of Computer Music*, The MIT Press, Cambridge, Massachusetts, 1969.

McCalla, T.R., *Introduction to Numerical Methods and FORTRAN Programming*, Wiley, New York, 1967. A comprehensive and comprehensible survey of numerical methods. Recommended.

McCracken, D.D., and W.S. Dorn, *Numerical Methods and FORTRAN Programming*, Wiley, New York, 1964. An excellent text; careful treatment of errors.

"Medium-Scale Computer Survey," *Data System News*, June 1970, p.20.

Minter, J., and B. Lawrence, *Management Information Systems: Their Development and Use in the Administration of Higher Education*, Western Interstate Commission for Higher Education, Boulder, Colorado, 1969.

Moon, B.A.M., *Computer Programming for Science and Engineering*, Van Nostrand, Princeton, N.J., 1966. Contains many suggestions for computer work.

Post, D.L., *The Computer in Secondary School Math*, Entelek, Inc., Newburyport, Mass., 1969.

"Product Information Source Directory," *Data Systems News*, December, 1970, insert.

Reichardt, J. (ed.), *Cybernetic Serendipity*, Praeger, New York, 1969. A beautiful book with eye-opening examples of computer use.

Rosen, S., "Electronic Computers: A Historical Survey," *Computing Surveys*, Vol. 1, No.1, March 1969, pp.7-36.

Sammet, J.E., *Programming Languages: History and Fundamentals*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

Schwarz, G., O.M. Kromhout, and S. Edwards, "Computers in Physics Instruction," *Physics Today*, September 1969, p. 41. A survey.

Sedelow, S., "The Computer in the Humanities and Fine Arts," *Computing Surveys*, June 1970, p. 89. A scholarly review of the subject.

Sharpe, W.F., *The Economics of Computers*. Columbia University Press, 1969.

Sichery, S.C., "Topics in Numerical Analysis for High School Mathematics," *The Mathematics Teacher*, April 1970, p. 313. Useful classroom suggestions.

Silberman, H.F., and R.T. Filep, "Information Systems Applications in Education," *Annual Review of Information Science and Technology*, Vol. 3, Encyclopedia Britannica, Chicago, 1968, pp. 357-395.

Sippl, C.J., *Computer Dictionary and Handbook*, Bobbs-Merrill, Indianapolis, 1966. A useful reference work.

Smith, R.L., and C. Seagraves III, *A Study of the Role and Scope of Computers in the Texas System of Higher Education*, Coordinating Board, Texas College and University System, Austin, 1968.

Spencer, D.D., *Game Playing with Computers*, Spartan Books, 1969. An interesting approach with an almost universal appeal.

Stannard, C., "The Computer in General Physics Instruction," *Am. J. Phys. 38*, 1416 (1970).

Suppes, P., "The Use of Computers in Education," *Scientific American 215*, 207 (1966).

Sutherland, I.E., "Computer Displays," *Scientific American*, June 1970, p. 56. A well-written and nicely illustrated survey of computer graphics.

Tabbett, F.D., "Computers in Chemical Education," *Chemical and Engineering News*, January 19, 1970, p. 44. Indicates how computers can help to teach chemistry better.

Te Selle, D.W., "Pi, Polygons, and a Computer," *The Mathematics Teacher*, February 1970, p. 128. An interesting problem application.

"The Computer in Education," *The Bulletin of the National Association of Secondary School Principals*, February 1970, entire issue. A very useful collection of articles on the subject.

"The Mini Explosion," *Data System News*, February 1970, p. 9. A reasonably complete survey.

Thwaites, B., "The Role of the Computer in School Mathematics," *Educational Studies in Mathematics*, December-January 1969-70, p. 346. A very complete survey of the situation.

"Time-Sharing," *Datamation*, August 1969, entire issue.

Weiss, E.A. (ed.), *Computer Usage Fundamentals*, McGraw-Hill, New York, 1969.

White, P.T., "Behold the Computer Revolution," *National Geographic*, November 1970, p. 593. A well written and attractively illustrated article of general coverage and interest.

Withington, F.G., *The Real Computer: Its Influence, Uses, and Effects*, Addison-Wesley, Reading, Massachusetts, 1969.

Zinn, K.L., *Final Report of Project CLUE*, Project CLUE, Center for Research on Learning and Teaching, Ann Arbor, Michigan, 1970.

Zinn, K.L., *A Basic Reference Shelf on Interactive Use of Computers for Instruction*, ERIC Clearinghouse on Educational Media and Technology, Stanford University, 1968.

Zoet, C.J., "Computers in Mathematics Education," *The Mathematics Teacher*, November 1969, p.563

Appendix D.
Participants and Contributors

Edward Adams
IBM - Watson Research Center
Box 218
Yorktown Heights, New York 10598

David S. Adorno
Department of Mathematics
Ithaca College
Ithaca, New York 14850

Ross Amans
Department of Physics
Beloit College
Beloit, Wisconsin 53511

Willard N. Ander
Cullen College of Engineering
University of Houston
Houston, Texas 77004

Larry Anderson
Department of Mathematics
Wheaton College
Wheaton, Illinois 60187

Clifford D. Anger
University of Calgary
Calgary, Alberta
CANADA

Andrew T. Armstrong
University of Texas
Arlington, Texas 76010

Winifred Asprey
Department of Mathematics
Vassar College
Poughkeepsie, New York 12601

Joseph L. Aubel
Department of Physics
University of South Florida
Tampa, Florida 33620

Harry Auchter
Carroll College
140 Tenny Avenue
Waukesha, Wisconsin 53186

Leroy Baker
Department of Physics
Eastern Nazerene College
Quincy, Massachusetts 02038

Lawrence Balch
Wilbur Wright College
3500 N. Austin Avenue
Chicago, Illinois 60634

Howard Balfour
St. Xavier College
103rd Central Park
Chicago, Illinois 60655

Richard Ballard
2831 Ellsworth Street
Berkeley, California 94705

Richard J. Barton
University of Saskatchewan
Regina, Saskatchewan
CANADA

Pierre L. Bastien
Department of Physics
University of Washington
Seattle, Washington 98105

Mrs. Barbara T. Bauer
101 Michigan, Lot 23
Lawrence, Kansas 66044

Charles Bauer
Illinois Institute of Technology
Information Science Center
Chicago, Illinois 60616

Paul M. Bauer
101 Michigan, Lot 23
Lawrence, Kansas 66044

David R. Beaucage
Computer Science Department
Rutgers University
New Brunswick, New Jersey 08903

H. Jan Beajon
Department of Information Science
University of North Carolina
Chapel Hill, North Carolina 27514

Carol D. Bennett
Computer-Based Instruction Laboratory
University of Illinois
Urbana, Illinois 61801

Richard Benson
Seattle Community College
Seattle, Washington 98122

Edgar Bertaut
Federal City College
425–2nd Street, N.W.
Washington, D. C. 20001

Manjit Bhatia
Department of Physics
Bowie State College
Bowie, Maryland 20715

O. M. Bilaniuk
Department of Physics
Swarthmore College
Swarthmore, Pennsylvania 19081

Bernhard Binder
Department of Chemistry
Southern Oregon College
Ashland, Oregon 97520

Robert W. Bird
9201 S. W. 80 Terrace
Miami, Florida 33143

Donald L. Bitzer
Engineering Research Laboratory
University of Illinois
Urbana, Illinois 61801

Paul J. Black
Department of Physics
Birmingham University
Birmingham, ENGLAND

R. W. Blomme
Computer Based Instruction Laboratory
University of Illinois
Urbana, Illinois 61801

Ronald Blum
Commission on College Physics
University of Maryland
4321 Hartwick Road
College Park, Maryland 20740

Alfred M. Bork
Department of Physics
University of California-Irvine
Irvine, California 92664

William A. Bowers
147 Linden Street
Berlin, Massachusetts 01503

Lynn Bowles
N.C.E.C.S.
P. O. Box 12175
Research Triangle Park, North Carolina

Stephen S. Boyce
Berea College
305 Prospect Street
Berea, Kentucky 40403

John W. Boyd
Department of Physics
Knox College
Galesburg, Illinois 61401

Ogden G. Brandt
Department of Physics
St. Lawrence University
Canton, New York 13617

David C. Braught
Illinois Wesleyan University
Bloomington, Illinois 61701

Alan L. Breitler
13602 Grenoble Drive
Rockville, Maryland 20853

Brian Brennan
Department of Physics
Clarkson College of Technology
Main Street
Potsdam, New York 13676

H. R. Brewer
Georgia Institute of Technology
Atlanta, Georgia 30332

Thomas J. Brieske
Department of Mathematics
Georgia State University
Atlanta, Georgia 30303

George Brooks
7059 Jay Drive
Elyria, Ohio 44035

Robert M. Brown
1801 Covewood Drive
Huntsville, Alabama 35801

Walter E. Brown
Lawrence University
Appleton, Wisconsin 54911

Gerald Buck
414 William Street
Racine, Wisconsin 53402

Elliot L. Buell
Worcester Polytechnic Institute
Worcester, Maine 01609

C. Victor Bunderson
Computer-Assisted Instruction Lab.
University of Texas
Austin, Texas 78712

Norman A. Burandt
Pan American College
Edinburg, Texas 78539

R. A. Burnstein
Department of Physics
Illinois Institute of Technology
Chicago, Illinois 60616

Alex F. Burr
Department of Physics, Box 3D
New Mexico State University
Las Cruces, New Mexico 88001

Hugh W. Byfield
Department of Physics
Stevens Institute of Technology
Hoboken, New Jersey 07030

Wendell Caley, Head
Department of Physics
Eastern Nazarene College
Quincy, Massachusetts 02038

Lynell Cannell
252 Engineering Research Laboratory
University of Illinois
Urbana, Illinois 61801

Lowell Cannon
Cuyahoga Community College
2007 East Arndale Road
Stow, Ohio 44224

Kermit H. Carlson
Valparaiso University
Valparaiso, Indiana 46383

Thomas Carr
University of Chicago
Chicago, Illinois 60637

Yvan Chasse
Department of Physics
Universite Laval
Quebec 10, P.Q.
CANADA

Monroe Cheney
1217 West Cedar
Arlington, Texas 76010

Harry R. Chesley
327 North Taylor
Oak Park, Illinois 60302

Peter Chiarulli
Department of Physics
Illinois Institute of Technology
Chicago, Illinois 60616

Norman Chonacky
Department of Physics
Southern Connecticut State College
New Haven, Connecticut 06515

Sister M. Vera Clarke
Clarke College
Dubuque, Iowa 52001

LeRoy Colquitt
Institute for Services to Education
2001 S Street, N.W.
Washington, D. C. 20024

James B. Conklin, Jr.
Department of Physics
University of Florida
Gainesville, Florida 32601

Richard Conklin
Hanover College
P. O. Box 24
Hanover, Indiana 47243

David M. Cook
Department of Physics
Lawrence University
Appleton, Wisconsin 54911

James M. Cook
Middle Tennessee State University
Box 138
Murfreesboro, Tennessee 37130

Raymond Cowan
Department of Physics
Franklin College
Franklin, Indiana 46131

Jeremiah A. Cronin
Brown University
Providence, Rhode Island 02912

H. R. Crouch
Division of Natural Sciences
New College
Sarasota, Florida 33578

Stephen L. Cunningham
Behlen Laboratory of Physics
University of Nebraska
Lincoln, Nebraska 68508

Mr. & Mrs. John E. Curtis
Box 566
East Moriches, New York 11940

Bruce R. Danner
Department of Physics
Rose Polytechnic Institute
Terre Haute, Indiana 47803

Stuart L. Dattner
Department of Physics
University of Texas
Austin, Texas 78712

Michael DeCelles
University Du Quebec
2525 Laurier Boulevard
Quebec City, Quebec
CANADA

Joseph R. Denk
N.C.E.C.S.
P. O. Box 12175
Research Triangle Park, N.C. 27709

Robert Dentzman, O.F.M.
Quincy College
Quincy, Illinois 62301

Robert E. DesMaisons
Aiken Computation Center
33 Oxford Street
Cambridge, Massachusetts 02138

Robert J. Dirkman
Aiken Computation Center
33 Oxford Street
Cambridge, Massachusetts 02138

Evelyn S. Dow
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

Homer Easley
1140 Laurel Drive
Aurora, Illinois 60506

John Eastmond
Department of Physics
Brigham Young University
Provo, Utah 84601

Mark Ebersole
Computer Center
Oregon State University
Corvallis, Oregon 97331

Robert Ehrlich
Department of Physics
State University College
New Paltz, New York

H. Margaret Elliot
Department of Mathematics
University of Bridgeport
Bridgeport, Connecticut 06602

Gerald L. Engel
Hampden-Sydney College
Box 11
Hampden-Sydney, Virginia 23943

Carl Engleman
The MITRE Corporation
Bedford, Massachusetts 01730

D. J. Fader
370 Belgrave Avenue
Apartment 31
London, Ontario
CANADA

Richard T. Fallon
Defiance College
Defiance, Ohio 43512

Huoo-Long Fann
65 Franklin Street
Lawrence, Massachusetts 01840

Paul F. Feldker
3400 Pershall Road
St. Louis, Missouri 63135

F. Fenner
300 Plaza
Park Forest, Illinois 60466

Harold W. Fenrick
675 Union Street
Platteville, Wisconsin 53818

Robert Feretich
11326 South Neenah
Worth, Illinois 60482

LeBaron O. Ferguson
124 Cottage Street
New Bedford, Massachusetts

G. Field
Department of Mathematics
McMaster University
Hamilton, Ontario
CANADA

Arthur Fink
c/o Marge Gard
3507 Sunset Drive
Madison, Wisconsin

Edward J. Finn
Department of Physics
Georgetown University
Washington, D. C. 20007

Kenneth W. Ford*
3541 Arizona Street
Los Alamos, New Mexico 87544

Bruce Foster
1106 Hillcrest Drive
Denton, Texas

John M. Fowler, Director
Commission on College Physics
University of Maryland
4321 Hartwick Road
College Park, Maryland 20740

Sister Mary Eleanor Fox, SND
1601 Dixie Highway
Covington, Kentucky 41011

Burton D. Fried
Department of Physics
University of California
Los Angeles, California 90024

Earl W. Friesen
Department of Physics
San Francisco State College
San Francisco, California 94132

Sister Ignatia Frye
Marygrove College
Detroit, Michigan 48221

Robert G. Fuller
Department of Physics
University of Nebraska
Lincoln, Nebraska 68508

H. E. Galbreath
P. O. Box 1816
Chicago, Illinois 60690

Rober G. Garst
16422 Greenwood Avenue
South Holland, Illinois 60473

*On leave from the University of Massachusetts, Boston.

Ian R. Gatland
Georgia Institute of Technology
Atlanta, Georgia 30332

Laurie Gauvin
Universite Laval
Quebec 10, P.Q.
CANADA

J. D. Gavenda
Department of Physics
University of Texas
Austin, Texas 78712

John Gering
Kenosha Technical Institute
Kenosha, Wisconsin 53140

W. Edward Gettys
Department of Physics
Clemson University
Clemson, South Carolina 29631

Fred Giessow
19 Spoede Lane
St. Louis, Missouri 63141

Terry L. Gildea
Hewlett-Packard
P.O. Box 301
Loveland, Colorado 80537

Harold Glander
Carroll College
Waukesha, Wisconsin 53186

Leon Goldberg
Science Department
Glassboro State College
Glassboro, New Jersey 07305

Richard S. Goldhor
Computer Based Education Research Laboratory
University of Illinois
Urbana, Illinois 61801

Richard Gonzales
7223 Ontario
Hammond, Indiana 46323

John Goodman
Harvey Mudd College
Claremont, California 91711

Bryan Gore
Department of Physics
Central Washington State College
Ellensburg, Washington 98926

Thomas Gorka
New Hampshire Vocational
and Technical College
Claremont, New Hampshire 03743

Mary F. Gourley
Department of Physics
Austin College
Sherman, Texas 75090

Robrick Granhold
4952 North Bernard
Chicago, Illinois 60625

Willard L. Graves
John Hopkins University
Charles and 34th Street
Baltimore, Maryland 21218

Thomas B. Greenslade, Jr.
Department of Physics
Kenyon College
Gambier, Ohio 43022

David T. Grimsrud
2449 Livingston Street
Allentown, Pennsylvania 18104

I. C. Gupta
Department of Physics
University of Saskatchewan
Saskatoon, Saskatchewan
CANADA

Michael A. Hall, Director
Social Sciences, Inst. Programming Project
Beloit College
Beloit, Wisconsin 53511

Norman T. Hamilton
Computer Center
University of Illinois
Urbana, Illinois 61801

D. L. Hardcastle
Department of Physics
Baylor University
Waco, Texas 76703

D. N. Hansen
Florida State University
Tallahassee, Florida 32306

Douglas J. Harke
Department of Physics
State University College
Geneseo, New York 14454

Fred H. Harris, Director
Computation Center
University of Chicago
5640 South Ellis Avenue
Chicago, Illinois 60637

Ritchy Haynes
Cleary College
Ypsilanti, Michigan 48197

E. Keith Hege
Department of Physics
Hollins College
Hollins College, Virginia 24020

Carl E. Heilman
Pennsylvania Dept. of Education
592 West Main Street
Mt. Joy, Pennsylvania 17552

Maury P. Hepner
Aiken Computation Center
33 Oxford Street
Cambridge, Massachusetts 02138

Bruce Hess
Box 333
Taylor University
Upland, Indiana 46989

John C. Hiebert
Cyclotron Institute
Texas A & M University
College Station, Texas 77843

E. R. Higgins
University of Texas
Austin, Texas 78712

Dwight D. Hinds
Washtenaw Community College
P. O. Box 345
Ann Arbor, Michigan 48107

Iraj Hirmanpour
Embry-Riddle Aero University
P. O. Box 2411
Daytona Beach, Florida 32014

Paul O. Hoffman
6 Park Avenue
Maplewood, New Jersey 07040

Martin O. Holoien
Moorhead State College
Moorhead, Minnesota 56560

Robert W. Holz
Leeward Community College
Pearl City, Hawaii

F. R. A. Hopgood
Atlas Computer Laboratory
Chilton
Didcot, Berkshire
ENGLAND

Michael V. Howett
Route 1
Batesville, Texas 78829

Paul Hruboveak
2112 Oriole Street
McKeesport, Pennsylvania 15132

Mr. Maion Jay Hubers
Iowa Lakes Community College
308 North Nineteenth Street
Estherville, Iowa 51334

E. R. Huggins
Wilder Hall
Dartmouth College
Hanover, New Hampshire 03755

Reverend D. R. Hutchinson
Loras College
Dubuque, Iowa 52001

Edwin L. Iloff
Department of Physics
Sacramento State College
Sacramento, California 95819

Hillar Ilves
Franklin Pierce College
Rindge, New Hampshire 03461

Roland Jackson, III
Tuskegee Institute
P. O. Box 462
Tuskegee Institute, Alabama 36088

Jeffrey S. Jalbert
Denison University
6 Somson Place
Granville, Ohio 43023

Benjamin Joe
Chicago City College
2500 North Lake View Avenue
Chicago, Illinois 60614

David C. Johnson
1425 2nd Street, S. W.
New Brighton, Minnesota 55112

George Johnson III
Department of Physics
University of Dayton
Dayton, Ohio 45409

Herbert W. Jones
Department of Physics
Florida A & M University
Tallahassee, Florida 32307

C. S. Kalman
Loyola of Montreal
7141 Sherbrooke Street, West
Montreal 262, Quebec
CANADA

Mr. Jack A. Kamis
Middlesex County College
434 West 120th Street - 6D
New York, New York 10027

Ik-Ju Kang
Southern Illinois University
655 Notre Dame Avenue
Edwardsville, Illinois 62025

Paul Kaplin
Department of Physics
Beloit College
Beloit, Wisconsin 53511

John Karl
Wisconsin State University
Oshkosh, Wisconsin

Meyer Katzper
Faculty of Science and Math
SUNY
Plattsburgh, New York 12901

Hoyt McCoy Kaylor
939 9th Court West
Birmingham, Alabama 35204

Charles F. Keleman
143 Chestnut Street
Apartment C-21
Ithaca, New York 14850

Tim G. Kelley
Department of Phy s
Oregon State University
Corvallis, Oregon 97331

Alice J. Kelly
Department of Mathematics
University of Santa Clara
Santa Clara, California 95050

Francis R. Kendziorski
Western Connecticut State College
37 Fairview Avenue
Danbury, Connecticut 06810

Brother Stephen Kietzman
7100 Shore Road
Brooklyn, New York 11209

Clark H. Kimberling
Illinois Institute of Technology
Chicago, Illinois 60616

David Klahr
Carnegie Mellon University
Schenley Park
Pittsburgh, Pennsylvania 15213

Charles Korn
Middlesex County College
26 Wolff Avenue
East Brunswick, New Jersey 08817

Donald L. Kreider
Department of Mathematics
Dartmouth College
Hanover, New Hampshire 03755

Harvey Kriloff
University of Illinois
Chicago, Illinois

Cora M. Kromhout
CRICISAM
Department of Physics
Florida State University
Tallahassee, Florida 32306

Scott Krueger
Computer Center - Box 4348
University of Illinois
Chicago, Illinois 60680

Gregory Kulas
2100 Church Street
Stevens Point, Wisconsin 54481

Robert G. Kuller
1102 South 4th Street
St. Charles, Illinois 60174

J. Claude Ladouceur, Director
Computer Center
College Militaire Royal
St. Jean, Quebec
CANADA

Walter Laffer
Department of Mathematics
Chicago State College
Chicago, Illinois 60621

David F. Laird
Cincinnati County Day School
6905 Given Road
Cincinnati, Ohio 45243

Robert E. Lake
Pratt Institute
Brooklyn, New York 11205

Edward D. Lambe
Instructional Resources Center
State University of New York
Stony Brook, New York 11790

Robert Lambert
1413 West 4th Street
Williamsport, Pennsylvania 17701

Arlene S. Larsen
OSC-B-11, Upson Hall
Cornell University
Ithaca, New York 14850

Karl D. Larsen
College of Arts and Sciences
University of Bridgeport
Bridgeport, Connecticut 06602

Emmett M. Larson
424 Johnson Avenue
Apartment 17
Cape Canaveral, Florida 32920

Kurt F. Lauckner
Department of Mathematics
Eastern Michigan University
Ypsilanti, Michigan 48197

Robert G. Larson
Seneca College of Applied Arts & Tech.
1750 Finch Avenue East
Toronto, Ontario
CANADA

Sanford Leestma
Department of Mathematics
Calvin College
Grand Rapids, Michigan 49506

R. Bruce Lenam
Computation Center
Polya Hall
Stanford University
Stanford, California 94305

Thomas T. Liao
Engineering Concepts Curriculum Project
Polytechnic Institute of Brooklyn
Brooklyn, New York 11201

Robert E. Lindsay
IBM-Instructional Systems Development
Watson Research Center
Yorktown Heights, New York 10598

Betsy R. Little
N.C.E.C.S.
P. O. Box 12175
Research Triangle Park, N.C. 27709

Arthur W. Luehrmann
Department of Physics
Darmouth College
Hanover, New Hampshire 03755

Peter G. Lykos
Information Sciences Center
Illinois Institute of Technology
Chicago, Illinois 60616

John Magerlein
University of Michigan
Ann Arbor, Michigan 48104

Judy Malkin
Hogg Building 10
University of Texas
Austin, Texas 78712

Kenneth Mann
Department of Physics
University of Missouri
Columbia, Missouri 65201

Robert L. Marcus
Department of Mathematics
Central State University
Wilberforce, Ohio 45384

David Marker
Department of Physics - Room 127
Michigan State University
East Lansing, Michigan 48823

David A. Marsh
State University of New York
Box 401
Cobleskill, New York 12043

Donald C. Martin
Department of Chemical Engineering
North Carolina State University
Raleigh, North Carolina 27607

Hugh Martin
Indiana University
Terre Haute, Indiana 47809

David McCandlish
Department of Physics
Rochester Institute of Technology
One Lomb Memorial Drive
Rochester, New York 14623

Stephen F. McCormick
Computer Services - Harper E4
Claremont College
Claremont, California

Brother John McNamara
Manhattan College
Bronx, New York 10471

Samuel McNeary
Drexel University
Philadelphia, Pennsylvania 19104

John Merrill
Wilder Hall
Dartmouth College
Hanover, New Hampshire 03755

Peter J. Metropoulos
7 Julie Drive
Glenview, Illinois 60025

R. C. Mikkelson
Department of Physics
Macalester College
St. Paul, Minnesota 55101

A. P. Miller
Department of Physics
Brandon University
Brandon, Manitoba
CANADA

William F. Miller
SLAC
Stanford University
Stanford, California 94305

William Miller
1910 Mayflower Place
Goshen, Indiana 46526

Robert A. Mills
Department of Mathematics
Eastern Michigan University
Ypsilanti, Michigan 48197

Joan Mitchell
University of Illinois
252 Engineering Research Lab
Urbana, Illinois 61801

Andrew R. Molnar
Office of Computing Activities
National Science Foundation
Washington, D. C. 20250

Charles E. Monfort
2036 Briagate Lane
Kirkwood, Missouri 63122

Albert Montgomery
Franklin College
Franklin, Indiana 46131

Hugh L. Moore
Los Angeles Valley Community Coll.
Van Nuys, California 91401

Philip H. Moore
University of Hawaii
96-050 Farrington Highway
Pearl City, Hawaii 96782

Sam W. Morphew
Auburn University
138 Shelton Road
Auburn, Alabama 36830

Charles E. Moulton
Beaver College
327 Cricket Avenue
Glenside, Pennsylvania 19038

Linda T. Moulton
Beaver College
327 Cricket Avenue
Glenside, Pennsylvania 19038

Robert B. Muir
Department of Physics
University of North Carolina
Greensboro, North Carolina 27412

Charles Munch
University of California
Irvine, California 92650

Carl J. Naegele
2323 Linden Avenue
South Plainfield, New Jersey 07080

Dean Neher
210 East College Street
Bridgewater, Virginia

William J. Nellis
Department of Physics
Monmouth College
Monmouth, Illinois 61462

Herman Newstein
Drexel University
Philadelphia, Pennsylvania 19104

David Nichols
3 Gardner Terrace
Delmar, New York 12054

Robert Nichols
2286 Markham Avenue
Thousand Oaks, California 91360

Sherman B. Nornes
Department of Physics
Pacific Lutheran University
Tacoma, Washington 98447

Thomas L. Nunes
Department of Chemistry
State University College
New Paltz, New York 12561

Anthony G. Oettinger
Aiken Computation Laboratory
Harvard University
Cambridge, Massachusetts 02138

Jon M. Ogborn
Chelsea College
University of London
London, ENGLAND

Dan W. Olds
Wofford College
Spartanburg, South Carolina 29301

Mary Anne Overman
University of Florida
1524–K, N.W. 4th Avenue
Gainesville, Florida 32601

Edward Packel
Reed College
Portland, Oregon 97202

David Palmer
Department of Physics
Kalamazoo College
Kalamazoo, Michigan 49001

James H. Parry
University of Illinois
ERL Room 252
Urbana, Illinois 61820

George Payne
Wisconsin State University
Oshkosh, Wisconsin

Herbert D. Peckham
Gavilan College
5055 Santa Teresa Boulevard
Gilroy, California 95202

Anthony P. Peluso
3059 N. Kilbourn Avenue
Chicago, Illinois 60641

Bernadette Perham
Department of Mathematics
Chicago State College
Chicago, Illinois 60621

Irene A. Petz
Palomar College
San Marcos, California 92069

Sister Mary Raymond Pierotti
Mercyhurst College
East 38th Street
Erie, Pennsylvania 16501

Helen Pillans
Mills College
Oakland, California 94613

Jerome Pine
California Institute of Technology
Pasadena, California 91109

Andrew W. Plonsky
University of Scranton
Scranton, Pennsylvania 18510

William D. Ploughe
Ohio State University
174 West 18th Avenue
Columbus, Ohio 43210

Russell Poch
21 West 7th
Apartment 7
Fort Scott, Kansas 66701

Jim Porter
Department of Physics
Eastern Michigan University
Ypsilanti, Michigan 48197

John Powers
City Colleges of Chicago
Amundsen-Mayfair Campus
Chicago, Illinois 60630

John L. Prather
PMC Colleges
206 Sykes Lane
Wallingford, Pennsylvania 19086

Walter J. Pravica
611 North 6th Avenue
Maywood, Illinois 60153

Stan Rajnak
Kalamazoo College
1200 Academy
Kalamazoo, Michigan 49001

William Ralph
345 Lakewood Lane
Marquette, Michigan 49855

Mary Ranade
3140 South Michigan
Chicago, Illinois 60616

Gerald M. Rees
521 Scio Church Road
Ann Arbor, Michigan 48103

E. A. Remler
Department of Physics
College of William and Mary
Williamsburg, Virginia 23185

Jay Rhee
LaVerne College
LaVerne, California 91750

C. B. Richardson
Department of Physics
University of Arkansas
Fayetteville, Arkansas 72701

James Richardson
40 Harper Drive
Pittsford, New York 14534

Robert A. Roberts
Denison University
Granville, Ohio 43023

Reverend Edward F. Robinson, S.J.
Xavier University
Victory Parkway
Cincinnati, Ohio 45207

John W. Robson
Department of Physics
University of Arizona
Tucson, Arizona 85721

William A. Rogers*
College of Petroleum
Dhahran, SAUDI ARABIA

Peter Roll
School of Physics and Astronomy
University of Minnesota
Minneapolis, Minnesota 55455

Peter E. Roohr
5 West Las Vegas Street
Colorado Springs, Colorado 80903

Sema Marks Rosenbaum
Westinghouse Learning Corporation
100 Park Avenue
New York, New York 10017

Thomas Rossing
St. Olaf College
Northfield, Minnesota 55057

Robb N. Russell
Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19144

Sister M. Briant Ryder
Department of Physics
Clarke College
Dubuque, Iowa 52001

*U.S. address: Apartment 43
29 Benwood Avenue
Buffalo, New York 14214

Claude St. Pierre
Department of Physics
Universite Laval
Quebec 10, P.Q.
CANADA

Dr. Frank J. Sarno
Polytechnic Institute of Brooklyn
333 Jay Street
Brooklyn, New York

Vincent J. Sawsinski
1945 North 77th Court
Elmwood Park, Illinois 60635

Edward A. Saunders
Department of Physics
U. S. Military Academy
West Point, New York 10096

Donald A. Schaefer
Bettendorf High School
2816 Middle Road
Bettendorf, Iowa 52722

Harry M. Schey
Education Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Glenn Schmitz
18150 South Anthony Avenue
Country Club Hills, Illinois 60477

Guenter Schwartz, Acting Director
Center for Research in College
Instruction of Science and Math
Department of Physics
Florida State University
Tallahassee, Florida 32306

Judah L. Schwartz
Educational Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Robert L. Sells, Chairman
Department of Physics
State University College
Geneseo, New York 14454

A. T. Serlimitsos
Department of Physics
U. S. Naval Academy
Annapolis, Maryland 21402

Noah Sherman
Department of Physics
University of Michigan
Ann Arbor, Michigan 48104

Bruce A. Sherwood
Engineering Research Laboratory
University of Illinois
Urbana, Illinois 61801

H. Chang Shih
Roosevelt University
430 South Michigan
Chicago, Illinois 60605

D. Shirer
Department of Physics
Valparaiso University
Valparaiso, Indiana 46383

Jerry Silver
Department of Mathematics
Ohio State University
Columbus, Ohio 43210

R. Small
Department of Physics
University of Alberta
Edmonton 7, Alberta
CANADA

Igal Staw
Polytech Institute of Brooklyn
Brooklyn, New York 11201

Douglas Smith
Ohio State University
Columbus, Ohio 43210

Joseph A. Smith
6940 South Peoria
Chicago, Illinois

Larry R. Smith
15125 Farmington Road
Livonia, Michigan 48154

Evan S. Snyder
Ursinus College
80 Linfield Road
Collegeville, Pennsylvania 19426

John D. Spangler
Department of Physics
Kansas State University
Manhattan, Kansas 66502

Preston Stein
Department of Mathematics
York University
Toronto, Ontario
CANADA

Warren B. Stenberg
Department of Mathematics
University of Minnesota
Minneapolis, Minnesota 55455

Kenneth S. Stephens
LeTourneau College
Longview, Texas 75601

Raymond Steyer
Beloit College
Beloit, Wisconsin 53511

Philip E. Sticha
5230 Even Star Place
Columbia, Maryland 21043

Carl F. Stocker
Miami Dade Junior College
Miami, Florida 33167

Arturs A. Stomers
Villa Maria College
2551 West Lake Road
Erie, Pennsylvania 16505

Ronald E. Stoner
Department of Physics
Bowling Green State University
Bowling Green, Ohio 43403

Harold S. Story
Department of Physics
State University of New York
Albany, New York 12203

Peter A. Stoycheff
5600 S.W., U.S. 42
RD #2
Ostrander, Ohio 43061

Charles M. Strauss
Division of Applied Mathematics
Brown University
Providence, Rhode Island

Thomas D. Strickler
Berea College
114 Van Winkle Grove
Berea, Kentucky 40403

Eric Stusnick
Department of Physics
Niagara University
Niagara, New York 14109

David R. Swanson
Illinois Institute of Technology
Chicago, Illinois 60616

Robert E. Swartwout
Engineering Sciences Building
West Virginia University
Morgantown, West Virginia 26506

Leonard J. Swiatkowski
7522 North Neva Avenue
Niles, Illinois 60648

Fred Tabbutt
Department of Chemistry
Reed College
Portland, Oregon 97202

Steven L. Tanimoto
30 Shepard Street
Cambridge, Massachusetts 02138

Edwin F. Taylor
Education Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

M. Gene Taylor
Bloomsburg State College
Bloomsburg, Pennsylvania 17815

Wade J. Temple
Randolph Macon College
P. O. Box 661
Ashland, Virginia 23005

Paul Tenczar
Computer-Based Instruction Laboratory
University of Illinois
Champaign, Illinois

Susie Thomas
6941 South Crandon Street
Chicago, Illnois

Raymond E. Thompson
Educational Testing Service
Rosedale Road
Princeton, New Jersey 08540

Robert A. Todd
Onondaga Community College
700 East Water Street
Syracuse, New York 13210

Hubert Tolman
Northeast Louisiana University
Monroe, Louisiana 71201

Thomas Ullan
Department of Physics
University of Puerto Rico
Mayagüez, Puerto Rico 00708

Dale Underwood
Chicago State College
6800 South Stewart
Chicago, Illinois 60616

William Updegraff
707 North C Street
Indianola, Iowa 50125

Val R. Veirs
Illinois Institute of Technology
Chicago, Illinois 60616

Anton Vierling
EDINET–Honeywell Inc.
2701 4th Avenue, South
Minneapolis, Minnesota 55408.

Francis Vilmain
Department of Physics
University of Northern Iowa
Cedar Falls, Iowa 50713

N. J. Vora
Box 321
Hawkins, Texas 75765

Diane Wagner
Department of Mathematics
University of Michigan
Flint, Michigan

Duane Walker
Department of Mathematics
Lewis College
Lickport, Illinois

Robert A. Walking
University of Maine
96 Falmouth Street
Portland, Maine 04103

Robert Ward
University of Chicago
Chicago, Illinois 60637

Ulysses Victor Ward
Los Angeles Southwest College
11514 South Western Avenue
Los Angeles, California 90047

Donald Weiner
Department of Physics
Syracuse University
Syracuse, New York

Harold Weinstock
Department of Physics
Illinois Institute of Technology
Chicago, Illinois 60616

William D. Weir
Reed College
Portland, Oregon 97202

Bob Weiss
University of California
San Diego, California

Dennis Wellnitz
Gustavus Adolphus College
St. Peter, Minnesota 56802

Frederick Werner
Xavier University
Cincinnati, Ohio 45207

Amy West
Berkshire Community College
Pittsfield, Massachusetts 01201

Roscoe D. White
Department of Physics
University of California
Los Angeles, California 90024

Samuel Wiley
Department of Physics
California State College
Dominguez Hills, California 90247

James B. Wilkinson
Erskine College
Box 949
Due West, South Carolina 29639

Richard M. Willett
Department of Electrical Engineering
Iowa State University
Ames, Iowa 50010

Dale R. Winder
Department of Physics
Colorado State University
Ft. Collins, Colorado 80521

Betty Windham
218 Grand Canyon Parkway
Hoffman Estates, Illinois 60172

Gordon A. Wolfe
Department of Physics
Southern Oregon College
Ashland, Oregon 97520

Mrs. John H. Wolff
University of Maryland
College Park, Maryland 20742

Seymour J. Wolfson
Wayne State University
Detroit, Michigan 48202

Hugh Woodford
Brigham Young University
Provo, Utah 84601

Hugh Young
808 State Street
Osage, Iowa 50461

Karl L. Zinn
Center for Research on
Learning and Teaching
University of Michigan
Ann Arbor, Michigan 48104

Zolton Zomori
Department of Physics
McMaster University
Hamilton, Ontario
CANADA

Eric Zornberg
Wellesley College
Wellesley, Massachusetts 02181