DOCUMENT RESUME

ED 027 219 SE 006 294

Proceedings of the Meeting on Computer Science in Electrical Engineering of the Commission on Engineering Education, Stanford University, October 24-25, 1968.

Commission on Engineering Education, Washington, D.C.

Pub Date Oct 68

Note-20p.

EDRS Price MF-$0.25 HC-$1.10

Descriptors-*College Science, *Computers, *Conference Reports, Electricity, Engineering, *Engineering Education, Workshops

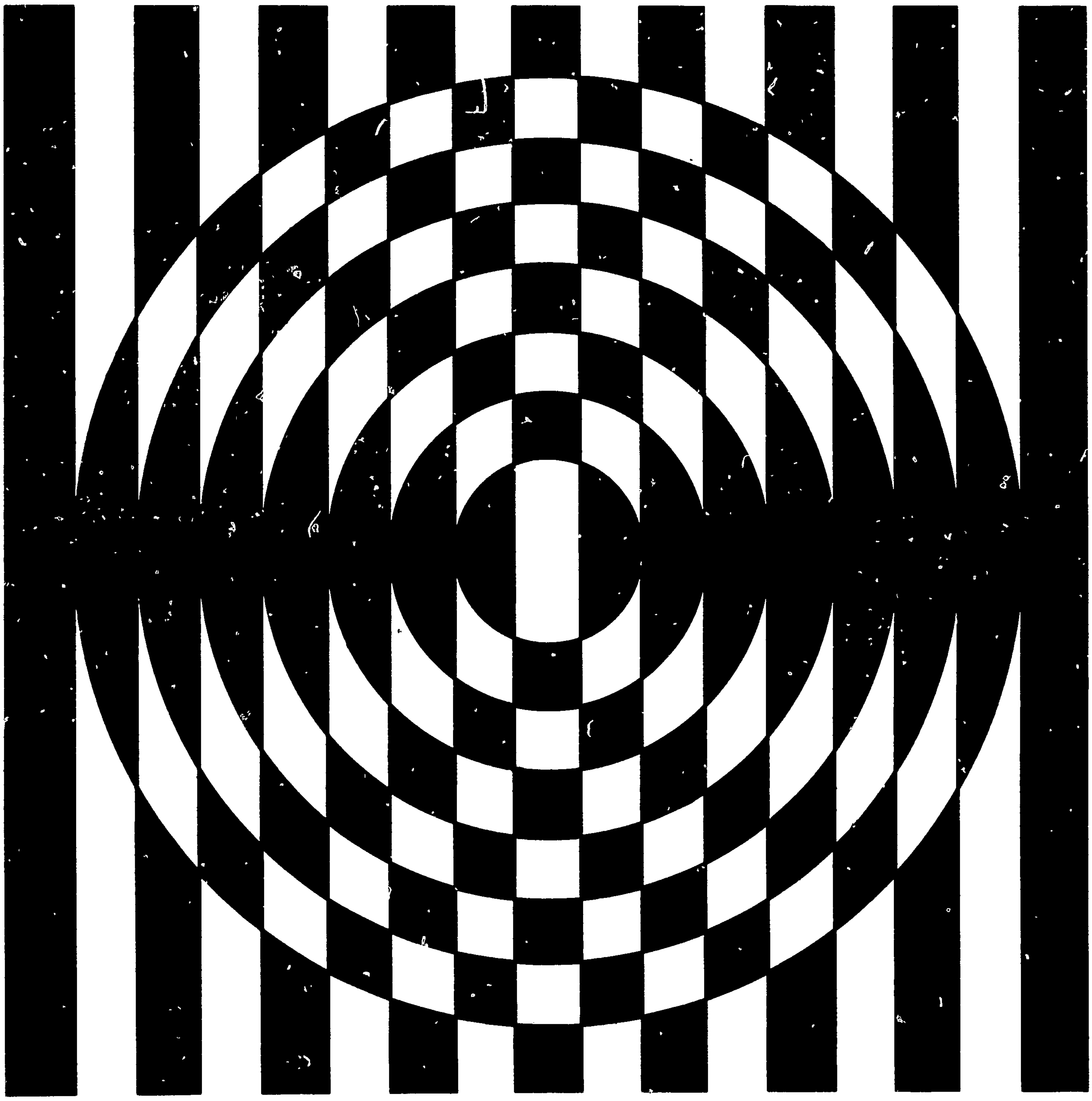
Identifiers-Commission on Engineering Education, National Academy of Engineering

This report contains the proceedings of a professional meeting on computer science held at Stanford University in October, 1968. Three papers presented at this meeting are published in the report. C. L. Coates spoke on "University Education in Computer Engineering"; L. P. Huelsman spoke on "Digital Computer Applications at the University of Arizona"; and Paul Ely and Robert Brunner spoke on "Requirements for Engineering Education from Hewlett Packard's Point of View." Three workshops on the use of computers in engineering are also discussed. A detailed meeting agenda and a list of conference participants are included. (BC)

PROCEEDINGS

Meeting on Computer Science in Electrical Engineering of the Commission on Engineering Education

STANFORD UNIVERSITY **OCTOBER 24-25, 1968**



PROCEEDINGS OF THE MEETING ON COMPUTER SCIENCE IN ELECTRICAL ENGINEERING

October 24-25, 1968

Stanford University

Stanford, California

Sponsored by the

COSINE Committee

Commission on Engineering Education

of the

National Academy of Engineering

2101 Constitution Avenue N.W.

Washington, D.C. 20418

2/3

TABLE OF CONTENTS

I. INTRODUCTION

The meeting held at Stanford on October 24 and 25 was the third meeting intended for department heads or their representatives to be sponsored by the COSINE Committee. The object of this meeting was to report on progress that was being made to incorporate the computer and computer science in the electrical engineering curriculum, to study problems remaining, and to provide for a free exchange of information.

It became clear at the meeting that the problems remaining are operational rather than conceptual. The advantage of the computer in teaching traditional material, the necessity for teaching all students a minimum of material in computer science, and the desireability of incorporating within the electrical engineering program an option for the education of computer engineers -- these three objectives seem to be generally accepted. The tasks that remain are the preparation of suitable course outlines, the writing of suitable textbooks, the development of suitable hardware and software for the implementation of a computer orientation within the curriculum. Clearly, these are tasks that can best be accomplished working individually but sharing the results once they are available.

The Proceedings of this conference are intended to record some of the talks presented. Other information, such as the reports of the three task forces, may be obtained directly from the Commission on Engineering Education.

II. UNIVERSITY EDUCATION IN COMPUTER ENGINEERING

C. L. Coates
Computer Science and Electrical Engineering
The University of Texas at Austin

My purpose today is to examine university education in computers in order to determine the way that it is developing and the areas, if any, that it is neglecting. We will focus our attention on the organization of computer curricula and attempt to correlate the objectives which curricula fulfill with the requirements that these objectives impose upon the educational environment.

We will not be concerned with departmental jurisdictional considerations because these must be resolved at each institution in terms of the policies that govern academic organization. My presentation assumes an institutional policy based upon curricula objectives rather than upon subject identification when these two are in conflict, because this represents the policy of most universities. No real limitations are encountered under the opposite situation; however, when university policy demands a rigid partition of both curricula objectives and subject identification, then there are important educational programs which that university cannot provide.

During the course of this discussion, I shall attempt to establish the validity of the following hypotheses: (1) At the present time many of the universities of this country offer a computer science program on either the graduate or the undergraduate level; (2) The number of these is growing rapidly and most are tending more and more toward a science oriented education; (3) Education in computer engineering is being neglected at most institutions; (4) Such education is the responsibility of the college of engineering and requires an engineering educational environment.

Let us first consider the growth of computer education programs as determined by a survey conducted during the period from September 1966 to March 1967 and as summarized in Figure 1. Parenthetically, let me indicate, for those of you who believe these results are in serious error, that the data were taken from the Appendix of the Academic Press book entitled *University Education in Computing Science* that was edited by Aaron Fineman. Reference to this data will show that slightly different results from those shown in Figure 1 might be obtained by different interpretation; however, you will find that, although the details may differ slightly, the trend to which I shall refer will not be altered.

Figure 1 shows that there were 58 computer science degree programs available in 1964 and by 1968 this number was expected to have increased to 240. By comparison, there were 19 computer option programs in electrical engineering in 1964, and this number was expected to have increased to 23 by 1968. Moreover, there were approximately four programs during each period in the Miscellaneous Options category that were also in engineering. It is fairly clear from Figure 1 that computer education is associated with computer science and that very little is associated with engineering. *A priori* this is not necessarily bad. What is important, is adequate education in computers; and if the computer science programs provide what is needed, there is no need for programs within engineering. It is my contention, however, that this is not the case.

I contend that most computer science programs are directed toward the software and abstract theoretical aspects of computers. I contend that most computer science curricula are confined to the areas of programming, numerical analysis, formal languages, abstract automata theory, and certain research oriented application areas. Moreover, I contend that education is neglected in the hardware aspects of computers, in the hardware-software interface area, and in systems for which the computer is a component part. These are problems of engineering for which an engineering training is required. I contend that computer science cannot provide this education because of two fundamental limitations:

(1) The majority of the faculty of most computer science programs are products of an arts and sciences education; and, therefore, they do not have the knowledge, experience or interest that is necessary to provide an engineering oriented education.

(2) Most computer science programs are located in the arts and sciences college of the university; and, as a result, it is not possible for the student to obtain the necessary engineering background because of curriculum constraints imposed by the college or by the faculty advisory system.

Do these assertions have any basis in fact?

Consider the proposed ACM Undergraduate Curriculum that was published in the March, 1968 issue of the *Communi-*

Program	Number of Existing Programs 1964-65			Expected Number of Programs 1968-69		
	B.S.	M.S.	Ph.D.	B.S.	M.S.	Ph.D.
Computer Science	13	29	16	96	93	51
Business Data Processing	6	3	1	15	4	2
Option in Mathematics	7	8	6	14	12	7
Miscellaneous Options	10	13	9	16	19	14
Option in Electrical Engineering	5	8	6	7	9	7

Figure 1. Number of Computer Oriented Degree Programs

cations of the ACM and that is summarized in Figure 2. This shows 30 possible courses in the computer science area, and it is clear that few, if any, of these are engineering oriented. The 12 courses enclosed by the shaded region are the recommended core courses; and, certainly, none of these have any engineering orientation whatsoever. One could dwell on this at length, but in my opinion the figure sets forth more clearly than words that this is a computer science curriculum and not a curriculum for computer engineering. In no respect is it concerned with education in the hardware aspects of computers, with systems, or with hardware-software interface considerations; and rightly so. For it is a curriculum that was designed by a committee, most of whom have no interest, experience, or appreciation of engineering; for a student body of the arts and sciences college who will not have the opportunity to obtain an engineering background and who will be taught by a faculty whose academic credentials are not from engineering.

One might envision computer engineering education as part of the existing computer science programs, but I doubt that this is realistic with the university academic organizational structure that has been assumed. Consider for a moment Figure 3 which shows the four-year undergraduate arts and science college program at the University of Texas at Austin. A study of the curriculum shows that, although the undergraduate student has considerable freedom in the selection of courses, this freedom does not include courses offered by the College of Engineering. This may not be a typical situation, although I suspect that this type of curriculum constraint exists in the arts and sciences colleges of a number of universities. But irrespective of whether or not this type of formal constraint is prevalent, the fact remains that very few, if any, students from the arts and sciences colleges enroll in courses in engineering. In fact, do you know of any arts and sciences students for whom engineering is their major or even their minor subject area?

Please don't misunderstand me. I am not criticizing the ACM undergraduate curriculum for computer science, the organization of computer science education within the College of Arts and Sciences of the universities, or the curricula of these colleges. Rather, I am trying to show that as a result of these factors, computer science education is concerned with the science of computers and that education in computer engineering is being neglected. Moreover, it is my assertion that these factors prevent computer science programs from fulfilling the computer engineering need.

There is an ever-increasing demand for computer engineers on all degree levels. The utilization of general and special purpose digital systems, both as stand alone devices and as components of larger systems, demands engineers who are trained in the analysis, organization, and design of systems that perform one or more of the functions of control, communications, recognition, processing and retrieval. Education for this type of engineering requires a curriculum that provides a broad background in communications, controls, digital systems and optimization, as well as in mathematics and other closely related engineering subjects. This can only be accomplished when graduate and undergraduate programs are available in engineering that have been specifically designed for the purpose. This is particularly important with regard to undergraduate education because undergraduate programs usually have the most restrictive curriculum constraints.

Many of the topics that are fundamental to computer engineering education are already offered by the electrical engineering and the computer science departments of most institutions. What usually is not provided, however, is the flexibility that would permit concentrated study in this area and, thereby, provide the necessary incentive for the continued development of the program as well as provide engineers with the necessary training. As a result, most of today's engineering graduates are not qualified for work in computer engineering. Moreover, many engineering colleges are not completely prepared to offer such a program because of neglected course development in some of the essential subject areas.

Probably the most practical way of initiating computer engineering education, especially at the undergraduate level, is as an option program in electrical engineering. My purpose today is not to discuss the details of such a program since this is a topic that COSINE will study during the coming period with your help. Nevertheless, one can suggest that it might compare with the normal electrical engineering undergraduate program as shown in Figure 4. This figure indicates the relative emphasis of the major subject areas in the normal electrical engineering program and in the proposed computer engineering program.

To some of you the suggestions of Figure 4 may seem a radical departure from the normal electrical engineering curriculum and, in fact, may cause concern because some of the established fundamental areas are not given adequate weight. I would suggest, however, that what is considered fundamental in electrical engineering depends upon the time period being considered. In support of this I invite you to compare the undergraduate electrical engineering curricula of the late 1930's and early 1940's with the curricula of today.

I would suggest that the beginning of a new epoch is a transition period during which the demands on the educational program are excessive because it must satisfy the needs of the passing epoch and the needs of the new one. During such periods the only way to satisfy both demands is by program dichotimization. In support of these assertions I refer you to the transition period between the epoch concerned with education in electrical power and that concerned with electronics.

Finally, I would suggest that toward the end of an epoch the educational program does not directly serve those subjects that were dominant in the previous epoch nor those that are responsible for the emerging one. In support of this, consider Figure 5 which lists the technical journals currently published by the Institute of Electrical and Electronics Engineers. Since this is the professional organization for electrical engineering, the technical areas that its publications serve must represent those that comprise electrical engineering. Certainly, no undergraduate program today directly serves all of these.

It is my firm belief that electrical engineering education is again in a transition period. I believe that we are witnessing the beginning of a new epoch, which for the purpose of this talk, I have chosen to call computer engineering; I believe that education in computer engineering is different from education in

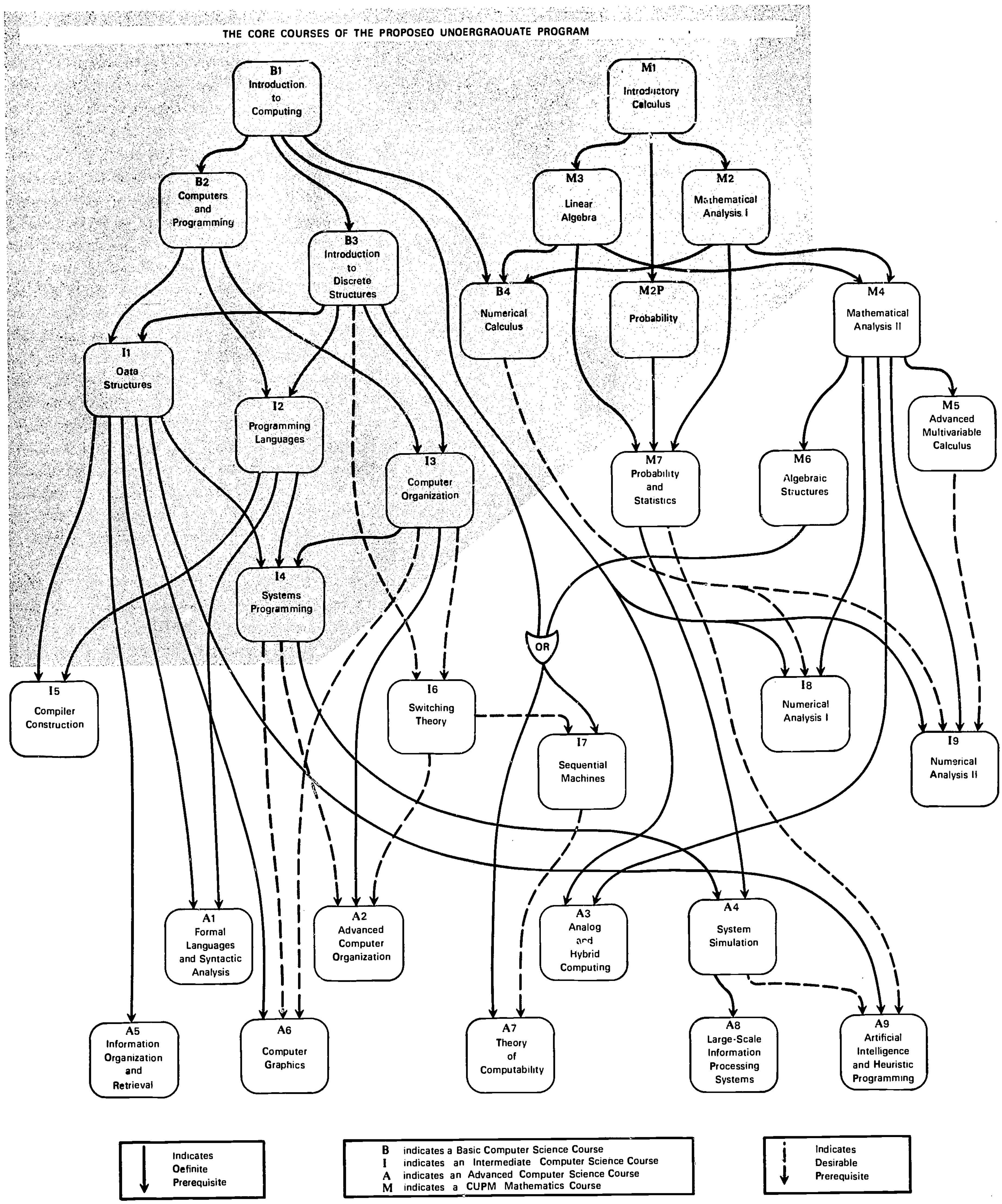


Figure 2. ACM Proposed Undergraduate Curriculum in Computer Science

Freshman Year

a) English 603

b) Biology 607

c) History 609

d) Courses 406 and 407 in a foreign language

e) Six semester hours in mathematics

f) If a student qualifies to take an elective, he has a choice of those listed under ELECTIVES

g) Required health and physical education

Sophomore Year

a) Physical Science: Chemistry 605 or 801 or Physics 609 or 801

b) Philosophy 610Q

c) Six semester hours of United States history or Government 610

d) Six semester hours in the foreign language taken in the freshman year

e) Elective, six semester hours of those listed under ELECTIVES

f) Required health and physical education

Junior Year

a) English 321 and three additional semester hours of advanced English

b) Six semester hours of advanced classical civilization

c) Tutorial Course 659, Special Studies

d) Six semester hours of United States history or Government 610

e) Elective, six semester hours selected from those listed under ELECTIVES

Senior Year

a) Tutorial Course, three semester hours selected from 660H, 379H or 679H

b) Elective courses sufficient to make a total of 120 semester hours selected from those listed under ELECTIVES

ELECTIVES: anthropology, classical civilization, economics, English, fine arts, foreign language, geography, geology, government, linguistics, mathematics, natural science, philosophy, psychology, sociology

Figure 3. The Undergraduate Curriculum of the College of Arts and Sciences of the University of Texas at Austin

Major Subject Area	Relative Emphasis	
	Normal Electrical Engineering	Computer Engineering
Mathematics	E	E
Electromagnetic Theory	E	L
Network Theory	E	L
Electron Materials and Devices	E	L
Electronic Circuits	E	L
Power Systems	L	—
Control Systems	L	E
Information and Communication Theory	L	E
Logic Design and Switching Theory	L	E
Machine Organization	—	E
Programming	L	E

Key: E denotes extensive emphasis

L denotes slight emphasis

Figure 4. Relative Emphasis of Subject Matter in Normal Electrical Engineering and Computer Engineering Undergraduate Programs

computer science and that both programs are essential to the development of the era that is emerging. I am not now suggesting that the electronics epoch is ending, although this may be true. I am suggesting, however, that we initiate the option system during this transistion period so that it is possible to adequately serve the needs of both as indicated in Figure 4.

Let me hasten to add that this is not the only way to provide adequate education in computers. In fact, it is my opinion that, ideally, computer science and computer engineering education should not be partitioned because of the academic organization of the university. Moreover, it need not be at those institutions where academic organization is determined by subject identification rather than by curricula objectives when these are in conflict. Nevertheless, such programs are few in number and it seems, therefore, that what I have proposed is the most practical way to fulfill a growing need.

In closing, if I might speak frankly, I would chide you as the leaders of electrical engineering education, as well as we of COSINE, for failing to recognize long ago the need for education in computer engineering. Our position is not defensible if we excuse ourselves on the jurisdictional grounds that computer science developed in arts and sciences. This is where it should be, as it is developing. Where we have failed, is to recognize that computer science education and computer engineering education are *not* the same and that there is a need for both. Our failure in this regard is difficult to explain because we are all aware of analogous situations. For example, that education in chemistry is not the same as education in chemical engineering or that education in biology is different from education in bio-engineering. Without much question, the need for computer engineers far exceeds the need for bio-engineers and maybe even the need for chemical engineers. It is not too late, gentlemen, but the time for action grows short. Thank you.

IEEE Transactions on

1. Audio and Electroacoustics
2. Aerospace and Electronic Systems
3. Antennas and Propagation
4. Automatic Control
5. Bio-medical Engineering
6. Broadcast and Television Receivers
7. Broadcasting
8. Circuit Theory
9. Communication Technology
10. Computers
11. Engineering Writing and Speech
12. Engineering Management
13. Electron Devices
14. Electromagnetic Compatibility
15. Electrical Insulation
16. Education
17. Industrial Electronics and Control Instrumentation
18. Geoscience Electronics

IEEE Transactions on

19. Industry and General Application
20. Instrumentation and Measurement
21. Information Theory
22. Magnetics
23. Man-Machine Systems
24. Nuclear Science
25. Microwave Theory and Techniques
26. Parts, Materials and Packaging
27. Power Apparatus and Systems
28. Reliability
29. Sonics and Ultrasonics
30. Systems Science and Cybernetics
31. Vehicular Technology
32. IEEE Journal of Quantum Electronics
33. IEEE Journal of Solid-State Circuits
34. IEEE Student Journal
35. IEEE Spectrum
36. Proceedings of the IEEE

Figure 5. Technical Publications of the Institute of Electrical and Electronics Engineers

III. DIGITAL COMPUTER APPLICATIONS AT THE UNIVERSITY OF ARIZONA

L. P. Huelsman
University of Arizona, Tucson

I. Introduction

One of the major advances that has occurred in the techniques of engineering in the previous decade has been the development of the computer and the application of its computing power to a myriad of engineering problems. A realistic evaluation of this "computer revolution," however, indicates that the progress which has occurred in applying computers to the solution of engineering problems has not always been matched by a corresponding progress in introducing computers into the undergraduate engineering curriculum. This paper will describe the efforts which have been initiated in the Department of Electrical Engineering at the University of Arizona to develop a program designed to make the undergraduate electrical engineering students aware of the impact of computers and to teach them the use of these devices. The paper also describes techniques whereby computational techniques may be introduced into undergraduate courses.

II. Learning to Use Computers

One of the ways *not* to introduce computational techniques to an undergraduate classroom is to give a course in Fortran. At first glance, the preceding statement seems almost paradoxical. However, such a procedure is similar to trying to teach a businessman, who never plans to visit Africa, a course in some obscure African bush language. Lacking the motivation for some future application, and barring the possibility that the businessman was a student of languages, it would not be expected that he would learn greatly from such an exposure. The same conclusion applies to teaching programming, and the use of some interpretative source language such as Fortran. Unless such teaching is closely tied to meaningful applications, the students will suffer greatly from lack of motivation, in which case the mechanics of programming are poorly learned and easily and quickly forgotten. Such a conclusion makes it imperative that programming, as such, *not* be taught by computer center personnel, or by a systems engineering department as a service course, but that it be taught directly by the engineering department which plans to make use of the techniques. In such an environment the learning of programming techniques and the application of them to meaningful problems becomes considerably easier to implement.

At the University of Arizona a course, EE-170, given in the second semester of the Sophomore year, has been prepared to accomplish the simultaneous teaching and application of programming. The goal of this course is to introduce the student to analog and digital computers, and at the same time to illustrate the application of computing techniques to situations which are defined in terms of engineering problems. The course is entitled "Introduction to Engineering Analysis I." It emphasizes the role of computers as tools to solve engineering problems. It concerns itself with the construction and analysis of models which may be applied to engineering systems. It demonstrates the use of computers to solve such models. Both digital and analog computers are introduced. With respect to digital computers, the use of peripheral machines such as the key punch, reproducing punch, listing machine, etc., is explained, and a thorough coverage of Fortran techniques including the development of the subroutine concept is included. The students are given practice in writing, submitting, and debugging their individual programs. With respect to analog computers, a laboratory facility is available in which the students may solve realistic problems on individual analog computers. In addition to teaching the theory of analog computation, considerable attention is paid to some of the particular difficulties which may be encountered in typical applications, for example, the scaling problem. In applying both digital and analog processes, meaningful treatments are made of both linear and nonlinear problems. Some numerical techniques are also introduced. At the end of this course the students have not only achieved a basic knowledge of how to use computers but they have also been exposed to the application of these techniques to meaningful problems typical of those they will encounter in later courses.

III. Application of Computers

Once a knowledge of some basic technique has been taught to a student, it is usually a great mistake to let even a single semester go by without building some more advanced knowledge on the basic technique which he has learned. This is one of the most fundamental principles of curriculum development. Such a principle is of even more importance in a subject such as the application of computational techniques, since, without immediate application, experience readily verifies the fact that such techniques can become rusty very rapidly. There are two basic types of application which may be incorporated in the curriculum. The first of these types may be labeled "forced" application. The term may be applied to courses which are included in the curriculum for the major purpose of providing such application and whose course content is chosen so as to accomplish this. A second type of application may be called "voluntary" application. It consists of the introduction of computer oriented techniques into traditional curriculum courses which are not inherently computer oriented. Examples of such would be courses in electronics, etc. Such applications are most important in that they illustrate most dramatically to the student to how great an extent the computer has become a part of all areas of engineering. An example of each of these two types of applications is given in the following paragraphs.

Forced Application of Computer Techniques

As an example of a course included in the curriculum for the specific purpose of increasing the students' proficiency in computational techniques, at the University of Arizona, a course EE-171, Introduction to Engineering Analysis II, has been set up. This course is scheduled for the first semester of the Junior year, thus it directly follows the course EE-170 described above. The major theme of this course is the simulation of dynamic systems through the use of digital and analog computers. Considerably more attention is given to the introduction of numerical techniques than was done in course EE-170. An introductory development and comparison of differential and difference equations is presented. A discussion of the limitations of various solution processes is made to give the student some feeling for the type of technique to be used in solving a given problem. For example, the numerical limitations of self-starting and multi-step integration rules are discussed, as are the effect of step size and order of different types of rules on numerical truncation. The choice as to whether to use analog or digital simulation is discussed. For example, one of the illustrative problems that has been used in the course is the simultation of a translunar orbit. This well-known nonlinear problem must be simulated digitally since its scaling range exceeds the limits of the analog computer. Another problem which has been used successfully is the classical pendulum problem. The analog simulation of this provides an illustration of the use of a Taylor series to represent the trigonometric term and also the use of implicit function generation to simulate such terms. Results of analog simulation can then be effectively compared with the use of such digital techniques as a 4th-order Runge-Kutta integration scheme. By the time the student has completed this course, he has developed a reasonable proficiency in the simulation of dynamic systems. Needless to say, he has considerably improved his basic programming knowledge through such applications.

Voluntary Application of Computer Techniques

The second type of application of basic programming knowledge was named voluntary application. In this type of application, an already existing course is modified to include digital computational techniques. As an example of such an application, at the University of Arizona in the first semester of the Junior year a course EE-122, Basic Circuit Theory, has been modified for the inclusion of various computer oriented topics. There are several major points which can serve as guidelines in preparing such a modification. The first of these is to determine which of the fundamental topics normally included in the course are logical candidates for computer implementation. For example, in a basic circuit theory course, topics such as integration, differential equation solution, magnitude and phase determination, solution of simultaneous equations, solution of equations with complex coefficients, etc., fall into this category. In general, the criteria for selecting such topics should include the following: the topics should reinforce the basic presentation of the theory given in the course; they should introduce new numerical techniques or review ones which have been previously covered; they should apply the knowledge of programming that the student has already developed; they should extend the range of the course material both in its application to larger scale problems and in its application to topics such as the time-varying or nonlinear cases which probably would not normally be included. The second major point to be observed in applying computer techniques to an existing course is the desirability of developing a software package containing subprograms applicable to the basic topics selected. Such a software package should emphasize the simplest practical implementation of the basic techniques. Such simplicity serves not only to minimize computer time but also makes it easier for the students to understand the operation of such programs. It is also desirable to make extensive use of subroutine format in preparing such a software package since this makes it easier for the students to cascade the separate elements of the package to achieve large program operation. Finally, it is most important to generate problems illustrating the application of the software package to typical problems in the course, to prepare adequate documentation describing the software package, and to validate the operation of the component programs. Some details on the implementation of these major points with respect to the basic circuit theory course referred to above are given in the next section.

IV. AN EXAMPLE OF VOLUNTARY APPLICATION

In this section we shall describe the implementation of the major points listed in the preceding section for a first course in circuit theory. First of all, let us consider some basic topics which are appropriate for such a course. A list follows:

1) Integration of Explicit Functions

Implementation of this topic by digital computational techniques permits the student to solve the integral relations for the terminal variables of inductors and capacitors for cases where explicit mathematical integration is not feasible.

2) Representation of a Piece-wise Linear Function

Implementation of this topic by digital computational techniques makes it possible for the student to consider the effect of solving the integral relations for the terminal variables of inductors and capacitors for the case where the applied excitations are not expressible in closed mathematical form.

3) Plotting

Implementation of this topic by digital computational techniques permits the student to express the results of numerical integration in visual form and also makes it possible for him to display solutions to many other problems.

4) Solution of Differential Equations

Implementation of this topic by digital computational techniques permits the student to solve the differential terminal relations for inductors and capacitors and also permits him to find a solution for the time-varying and nonlinear cases.

5) Solution of Matrix First Order Differential Equations

Implementation of this topic by digital computational techniques permits the student to solve the state equations in the time domain for an arbitrary class of networks.

6) Solution of Simultaneous Equations

Implementation of this topic by digital computational techniques permits the student to solve for the variables of resistive networks defined on a loop, node, or hybrid basis. Implementation of this topic using complex arithmetic makes it possible for the student to solve phasor problems.

7) Manipulation of Rational Functions

Implementation of this topic by digital computational techniques permits the student to differentiate polynomials, evaluate polynomials for specific complex values of their arguments, and make partial fraction expansions of rational functions. Thus, the student can readily perform inverse Laplace transformations.

8) Magnitude and Phase Determination

Implementation of this topic by digital computational techniques makes it possible for the student to construct magnitude and phase plots of various network functions and thus demonstrate the effects of changing the values of the network parameters.

The second major point discussed above for applying digital computational techniques in an undergraduate course is the development of a software package. For the topics which have been outlined above, such a software package might include the following.

1) A trapezoidal integration subroutine in which the user specifies the number of trapezoidal sections desired, thus giving him control over the accuracy of the operation.

2) A subroutine providing linear interpolation between a specified set of data points. This permits piece-wise linear representation of an arbitrary function.

3) Subroutines providing for different types of plots. For example, it is desirable to have the capability of making multiple plots as a function of some independent variable and also a capability for making "x-y" plots to demonstrate Nyquist plots and root-locus plots.

4) A scalar differential equation solving subroutine using a Runge-Kutta method and having provision for the user to supply the equations defining the problem.

5) A matrix differential equation solving subroutine similar to that used in (4), but with one-dimensional arrays to store the variables.

6) A Gauss-Jordan simultaneous equation solving subroutine. Two variations of this subroutine should be provided, one using real variables for solving resistor networks and a second using complex variables for solving phasor problems.

7) Subroutines for differentiating and evaluating polynomials and for making partial-fraction expansion of rational functions.

8) Subroutines for determining the magnitude and phase of a rational function.

It should be noted that in some of the elements of the software package described above, separate versions of the subroutines may be necessary depending on whether a Fortran II or IV compiler is available.

The third major point discussed above was the development of example problems, the generation of documentation for the software, and the validation of the operation of the component subroutines. In the example described here, the documentation started with the preparation of sheets describing the problems and the software. These loose sheets were then modified and expanded until they comprised a ditto set of notes which were given to the students. Finally, the notes have now achieved the form of a printed text complete with a solutions manual and accompanied by a pre-punched card deck containing the software for the course. The book was released in October, 1968 by the McGraw-Hill Book Company and is entitled "Digital Computations in Basic Circuit Theory."

One final detail concerning the implementation of the concepts described above is worthy of note as an aid to improving the efficiency with which these digital computational techniques are introduced, and thus making the success of such a program more certain. It is most helpful if the software package for the course can be added to the internal system library of the computer on which the programs are to be run. Such an implementation has been made at the University of Arizona on two different generations of computers, namely, an IBM-7072 which was used for the early versions of the software package and a CDC-6400 which is currently being used. Such an implementation can usually be accomplished very easily on most medium to large size computers. This procedure minimizes the size of the main programs which the students must submit and thus also minimizes the need for large amounts of tedious and error-prone key punching on their part.

V. CONCLUSION

The interest of the faculties of the majority of engineering colleges in the introduction of digital computational techniques into the undergraduate curricula is a very intense one. As an example of this interest, in the January 1968 issue of the Newsbrief published by the American Society for Engineering Education, an announcement appeared that this author had developed a set of notes for incorporating digital computational material in a basic circuit theory course. The response to this item was enormous. To date, over 150 inquiries have been received regarding the status of these notes and requesting a copy. The software package described in this paper has not only been used successfully in basic circuit theory courses but the instructors in several other undergraduate courses such as control, methods of engineering analysis, and advanced circuit theory, at the University of Arizona have begun making use of portions of the package. Such usage has been considerably encouraged by the ready availability of the software package as a part of the internal computer library. It is the plan of the Department of Electrical Engineering of the University of Arizona to extend digital computer applications of the type described in this paper to other undergraduate courses as soon as suitable software packages can be developed and documented.

IV. REQUIREMENTS FOR ENGINEERING EDUCATION FROM HEWLETT PACKARD'S POINT OF VIEW

by
Paul Ely, Robert Brunner
Hewlett Packard, Stanford, California

Paul Ely, manager of the microwave engineering group, first described the organization and work of engineers in his area. There are three basic types of working groups, the project team of four to ten engineers, the concept investigation team of one or two men, and the technology investigation team also composed of one or two men of complementary backgrounds. Typical work space for the engineer consists of his desk, his lab bench which is adjacent to his desk, and a time sharing console or computer nearby. The engineers are rapidly adapting to the use of computers in their everyday work.

With regard to what HP would like to see in the education of the young BS graduate, they feel there should be a solid base in mathematics, a broad base in physical science with some depth (real specialization can come on the job or in graduate school) and an increasing involvement in the use of computers both as an analysis tool and as a design tool. Their experience has been that a real engineering project in the senior year is excellent for motivating the student. They look for the well rounded BS student who also has some degree of social responsibility. They feel it very necessary to keep their top engineers continually exposed to as many new ideas as possible in the academic world.

Brunner stated that of their 1600 engineers, 900 are doing work like Paul Ely described with the other 700 in marketing, sales, etc. For all these engineers, especially the latter, they should know how to communicate effectively through reports, letters, talks, and participation in meetings. The engineers should also know what business is all about and its social implications. He must understand the prime motivation of the company. In this regard the new engineering employee is purposely involved right away in discussions about the new project he will be on. The discussions include the assets required for the job, the market possibilities of the product, the time and personnel required for completion of the project.

The value of summer jobs in industry for the undergraduate and graduate student was strongly stressed. HP employs one summer man for every five regular engineers. They would like to see some very effective mechanism set up for coupling industry to the engineering schools to develop summer jobs for the undergraduates. Brunner closed his talk stressing what they felt was required to develop fully the potential of each engineer. They see great value in a much closer and personal contact between enlightened faculty and the developing student.

V. WORKSHOP REPORTS

Workshop A: Use of Computers in Engineering Courses

Discussion Leader: S. Seely
COSINE Observer: W. H. Huggins
Attendance: 35

Owing to the short time available for discussion, this group was able to raise many more questions than it could answer. These questions are pertinent and important to decisions relating to the subsequent use of computers in engineering courses, and answers will be necessary before firm decisions can be made. Essentially, the concern revolves around the relation between costs and the educational importance and value of computers in engineering courses. Key points in the discussion were:

a. The extensive use of computers in engineering courses can materially increase the costs of education. How should schools raise the additional funds for the computer services?

b. Justification for additional funds for computer services will rest on an improved and a more efficient educational experience for the student. Is any data available on the relative effectiveness of computer based instruction? Can and should such data be collected?

In an endeavor to get at some of the implications of these questions, the following related matters were considered:

1. Can one compare the relative educational effectiveness of batch versus time-shared computers in education?

2. What is the relation of costs and effectiveness of small modern machines, e.g., the PDP-9 plus multiple terminals, versus the use of large central installations?

3. Should the average computing center seek to provide limited time-shared service by the use of a small computer, e.g., PDP-8, in an interface operation between teletype terminals and the central computer? (The U Mass system involves the PDP-8 and the CDC-3600).

4. A reasonable experience by the student in the use of computers for calculation, for simulation, and for modeling studies, is deemed to be most desirable. How extensive should such experience be, for optimum cost effectiveness?

5. Can one devise methods by which programmed materials, e.g., film strips, which are on ready call, can be displayed, rather than generating such materials on each occasion? Is this an economic procedure? What problems might be entailed in such an operation?

6. It is recognized that a computer orientation is probably of value in all courses. Would the general employment of an algorithmic organization of problems as the basis for discussion and solution of large problems serve to improve the quality of instruction? It would appear that such an approach would serve to remove inhibitions in undertaking problems that are more extensive than those possible by hand computation. Can such improvement be estimated quantitatively?

7. The availability of a computer in instruction offers the opportunity to employ simple examples to illustrate general

methods, e.g., the use of state variables in systems analysis, and in the development of general theorems. Is there a significant gain in carrying out extensive problems which require the actual employment of the computer?

The foregoing are fundamental questions that must be critically examined, and this workshop recommended that attention be directed to these and other related matters, since answers are needed by E.E. departments in securing and justifying funding for the use of computer time.

Workshop B: Computer Facilities and Software

Discussion Leader: W. H. Surber
COSINE Observer: J. F. Kaiser
Attendance: 14

The major topic discussed in this workshop concerned the type of computer facilities, both hardware and software, that should be available to effectively implement the development of computer-oriented engineering courses. The emphasis was primarily on general electrical engineering courses not in a specialized computer engineering option program.

The consideration of hardware facilities was relatively brief. The advantages of console access to the computing system with very rapid response times relative to the batch mode of operation are considerable, both for small problems when students are beginning to learn how to use the computer, and for large design-oriented problems when frequent interaction with the computer analysis program is desirable. The provision of an interactive computer capability permits an entirely different approach to many types of problems and would appear to be very important for future applications. The cost of providing such a facility may be a major hurdle for many institutions, however.

The provision of suitable software, i.e. special purpose library programs for various classes of problems, was considered at some length. The lack of such programs is one of the major difficulties faced by faculty members attempting to introduce computer applications in a realistic way as tools for the analysis and design of engineering systems. Three areas in which action might be taken to improve the distribution and the utility of the programs now available are the following:

(1) Information Exchange Concerning Program Availability
Some general purpose analysis programs, such as ECAP and various simulation programs, are quite well known and are available on many different types of machines. Many others have been developed, either completely or as major modifications to existing programs at various institutions, but have not been widely publicized. A list of such programs, their general capabilities, and information concerning their adaptability to various computer configurations would be very valuable. This might include such things as modifications of ECAP programs for graphical output or time-sharing operation, for example.

(2) Program Documentation
Better documentation for these programs in the form not only of user's manuals but also installation manuals, together with a discussion of the algorithms used, would be quite helpful.

(3) Educational Applications
A great deal of work has been done at a number of places concerning the best ways to make effective use of these computer programs in engineering courses. Very little of this has been published, however. Discussions of typical applications, including sample problems, might be very stimulating to faculty members considering modifications of their own courses.

A number of specific suggestions were made to encourage the wider exchange and applications of computer analysis and design programs. The possibility of publishing a preliminary program review article, perhaps comparing ECAP and JOBSHOP and discussing some applications, in the *IEEE Journal on Education* was discussed; plans have been made to do this. A computer program review section as a regular feature similar to a book review section might also be established in one or more of the IEEE journals. The encouragement of additional publications concerning educational applications of computers was discussed in some detail.

Workshop C: Computer Engineering Rather than Computer Science

Discussion Leader: C. L. Coates
COSINE Observers: J. B. Dennis, T. L. Booth
Attendance: 28

An initial survey of those in attendance showed that most Electrical Engineering Departments are using computers as a problem solving aid in both undergraduate and graduate courses. In addition, courses covering various aspects of digital circuit and system design as well as computer applications are offered by many departments.

About half of the attendees came from schools that have separate computer science departments while five came from schools that combine Electrical Engineering and Computer Science in the same department and eight came from schools with interdisciplinary programs. The general discussion showed that there does not, in most cases, seem to be any major problem in establishing cooperation between Computer Science and Electrical Engineering Departments. This is attributed to the fact that there are often joint appointments between departments and cooperation in working out courses which are common to both curriculums.

There seems to be a well defined separation of interest developing between the curricula of Computer Science Departments and Computer Science programs offered within Electrical Engineering Departments. The Electrical Engineering Departments are interested in the more pragmatic problems associated with the design and utilization of computers and information processing systems while the Computer Science Departments are interested in some of the more esoteric areas such as formal language theory and abstract automata theory.

The following general topic areas were identified as being of interest to an engineer working in the area of computers and information systems.

Mathematics	*Electrical Engineering*
Analysis	Circuits and Systems*
Modern Algebra	Electronics*
Formal Logic	Control, Communication and Information Theory*
Methods of Numerical Analysis	Solid State Electronics*
Transform Techniques*	Switching Theory and Logical Design
Combinatorics	
Statistics and Probability	

Computer Theory

Automata Theory
Formal Linguistics
Computability
Programming
Data Structures
Compiler Construction

A consensus was reached that the existence of a Computer Science Department must not interfere with the development of a strong computer oriented program in Electrical Engineering. Just as programs in chemistry and chemical engineering or solid state physics and solid state electronics exist jointly in many universities, there is a place and a need for Computer Science programs with both a science and an engineering orientation. Whether these programs will exist in one or two departments will probably depend upon local interests and historical precedents at each school. The field is still undergoing rapid developments and it is still too early to clearly identify any "best" structure for the way these programs should develop.

Whenever two departments exist every effort should be made to make it possible for students from one area to take courses in the other area without having to satisfy unreasonable prerequisite requirements. If the Electrical Engineering Department has assumed the overall responsibility for Computer Science offerings at a given university, then it becomes extremely important that these courses are available to a large group of non-Electrical Engineering students. This can be achieved by designing courses so that the specialized hardware and specific engineering oriented concepts are presented in parallel with the Computer Science courses that are common to both engineering and non-engineering students.

After discussing the general relationship between Computer Science and Computer Engineering, the group next considered the problem of what computer related concepts should be taught to all Electrical Engineering students and what material should be available in elective courses.

*These topics are not part of recommended ACM curriculum.

It seemed to be generally accepted that all students be required and able to use computers to solve problems in all areas of Electrical Engineering. A student, however, should not be required to develop a new program for each problem he is asked to solve. Instead a set of packaged programs such as ECAP, JOBSHOP, NET 1, should be available for the student's use. In this way the student can spend the major portion of his time concentrating upon the concepts being utilized to solve the problem rather than the unrelated problems associated with programming.

It was also realized that Electrical Engineers are using digital techniques in an ever increasing number of system design problems. Thus a student must be introduced to logical design and digital information processing techniques early in his undergraduate program so that he can judge whether digital or continuous techniques are best to solve any given problem. The minimum preparation would be one programming course and one course that emphasizes the concepts of logical design of digital networks and systems. A student after completing these courses, should be able to design simple combinational and sequential logic networks; he should understand how information is represented, stored and processed in digital form; and he should appreciate the interrelationship between the hardware and software capabilities of any given digital data processing system. Every effort should be made to have the student apply these concepts in a laboratory to the solution of realistic engineering problems. In this way he will have an opportunity to appreciate the interrelationship between the digital and the continuous concepts he encounters in his courses.

Electrical Engineering students who wish to do advanced work in the computer area should be able to take elective courses in such areas as programming systems design, machine organization, advanced switching and sequential machine theory, automata theory, numerical analysis, etc. The actual departmental locations of these courses will depend upon the circumstances at each school. In planning such programs, the student should be encouraged to develop sufficient breadth of understanding so that he can appreciate the interrelationships between the various disciplines that are important to the computer area.

Although there was a general consensus that computer related concepts should be an integral part of the Electrical Engineering curriculum, there was a general feeling that the actual realization of this goal involved the solution of several very important pragmatic problems. Some of these problems are:

a. Availability of textbooks with meaningful problems.

The faculty of most schools do not have the time or motivation to spend a large amount of time developing special programs or problems for each course they teach. There is an urgent need for texts which have been developed with the idea that the student will be expected to use a computer as a normal computational aid in solving the assigned problems.

b. Workshops for faculty.

A considerable amount of experimentation has been done and will be required to determine how various computer concepts should be brought into different courses. Regional workshops should be held so that the results of this work can be shared with faculty members from other schools.

c. Availability of software.

Packaged programs suitable for classroom use are often hard to find. Faculty members who have developed such programs should make them available by publishing them in a journal such as the IEEE Transactions on Education.

As a final recommendation the group suggested that the COSINE committee could collect and distribute information about activities going on at the different schools. Among the topics suggested were; information about current curricula in different schools, a definition of the topics which should be part of the core curriculum and those topics that should be available as elective topics, descriptive of new techniques and methods that have been developed to teach computer related courses, and an evaluation of the effectiveness of different approaches to teaching computer related topics.

APPENDIX A

Meeting Agenda

WEDNESDAY, OCTOBER 23, 1968

7:00 p.m. to 10:00 p.m. — Registration and Social Hour, Stanford Room, Rickey's Hyatt House

THURSDAY, OCTOBER 24, 1968, Room 270-1, Tresidder Memorial Union

Presiding: Edward J. McCluskey, Conference Chairman

9:00 a.m. — Welcome: Joseph M. Pettit, Dean of Engineering; John G. Linvill, Executive Head, Department of Electrical Engineering

9:30 a.m. — Introduction: COSINE Activities and Future Plans, M.E. Van Valkenburg

9:45 a.m. — Report on the Park City Conference on Computers in Undergraduate Education* W. H. Surber

10:15 a.m. — Coffee Break

10:30 a.m. — Computer and Information Systems Engineering, C. L. Coates

11:00 a.m. — Reports of Course Task Forces†
- #1—The First Course: M. E. Van Valkenburg
- #2—Computer Organization: E. J. McCluskey
- #3—Digital Circuits: J. Kaiser

12:00 Noon — Lunch: Outside, between Bowman Hall and Tresidder Mem. Union

Presiding: M. E. Van Valkenburg

1:30 p.m. — Reports: Some Present Programs Emphasizing Computer Applications or Computer Engineering
- Nick DeClaris, University of Maryland
- Lawrence P. Huelsman, University of Alabama
- Taylor Booth, University of Connecticut

3:15 p.m. — Coffee Break

3:30 p.m. to 5:00 p.m. — Workshops I

6:30 p.m. — No Host Cocktail Party in Rose Room, Rickey's Hyatt House

7:00 p.m. — Banquet in Rose Room, Rickey's Hyatt House
- Speaker: Dr. Edward Feigenbaum
- Title: Artificial Intelligence Research: 1968

8:30 p.m. to 10:00 p.m. — Workshops II, Rickey's Hyatt House

FRIDAY, OCTOBER 25, 1968, Room 270-1, Tresidder Memorial Union

Presiding: W. H. Surber

9:00 a.m. — Invited Address
- Introduction: Dr. Kay Magleby, Hewlett Packard
- Speakers: Paul Ely, Hewlett Packard; Bob Brunner, Hewlett Packard
- Title: "Requirements for Engineering Education from Hewlett Packard's Point of View"

10:00 a.m. — Workshop Reports and Recommendations, Workshop Chairmen

12:00 Noon — Adjourn

WORKSHOPS

Workshop A: **USE OF COMPUTERS IN ENGINEERING COURSES**
Tresidder Union, Room 270

Concepts, applications to realistic systems, use in design examples.

Workshop B: **COMPUTER FACILITIES AND SOFTWARE**
Bechtel International Center, Dining Room

Nature of the computer center, interactive console inputs, graphic output, on-line facilities. What software is now available? Future developments. Exchange of programs.

Workshop C: **COMPUTER ENGINEERING RATHER THAN COMPUTER SCIENCE**
Tresidder Union, Room 271

Distinctions between the two. Computer engineering or computer and information systems engineering?

*For information on the availability of this report, write Prof. W. Viavant, Department of Electrical Engineering, University of Utah, Salt Lake City, Utah 84112

†Available from Commission on Education of the National Academy of Engineering, 2101 Constitution Avenue, Washington, D. C. 20418

APPENDIX B – MEETING ATTENDEES

COMPUTER SCIENCE IN ELECTRICAL ENGINEERING

Meeting Sponsored by the COSINE Committee of the
Commission on Engineering Education
Stanford University, Stanford, California
October 24-25, 1968

E. P. Anderson
San Jose State College

H. A. Antosiewicz
University of Southern California

D. O. Akhurst
University of Arkansas

C. Beck
Tulane University

A. L. Betts
Washington State University

W. A. Blackwell
Virginia Polytechnic Institute

T. L. Booth
University of Connecticut

J. Bordogna
University of Pennsylvania

F. E. Brammer
Wayne State University

J. Cadzow
State University of New York

C. C. Carroll
Auburn University

R. J. Churchill
Colorado State University

C. L. Coates
University of Texas

R. J. Collins
University of Minnesota

R. L. Cosgriff
University of Kentucky

R. F. Cotellessa
Clarkson College of Technology

B. J. Dasher
Georgia Institute of Technology

J. B. Dennis
Massachusetts Institute of Technology

D. L. Dietmeyer
University of Wisconsin

N. DeClaris
University of Maryland

R. C. Dorf
University of Santa Clara

D. G. Dow
University of Washington

A. L. Duke
Tennessee Technological Institute

M. Ettenberg
City University of New York

D. C. Evans
University of Utah

A. G. Favret
Catholic University of America

F. C. Fitchen
South Dakota State University

D. C. Ford
John Wiley and Sons

H. Freeman
New York University

D. H. Gillott
Sacramento State College

R. D. Guyton
Mississippi State University

W. M. Hammond
Bradley University

J. C. Hancock
Purdue University

D. W. Healy
University of Rochester

E. W. Henry
University of Notre Dame

R. Heyborne
Utah State University

L. P. Huelsman
University of Arizona

W. H. Huggins
Johns Hopkins University

D. L. Johnson
Louisiana Polytechnic Institute

J. R. Johnson
Louisiana State University

J. J. Jonsson
Brigham Young University

J. F. Kaiser
Bell Telephone Laboratories

Z. Kaprielian
University of Southern California

M. Karnaugh
International Business Machines Corp.

W. W. Koepsel
Kansas State University

A. H. Koschmann
University of New Mexico

L. Kraus
Drexel Institute of Technology

T. S. Krile
Rose Polytechnic Institute

E. Kuh
University of California

F. Kuo
University of Hawaii

R. Lade
Marquette University

R. V. Langmuir
California Institute of Technology

J. LaPatra
University of California/Davis

J. Lehman
National Science Foundation

R. N. Linebarger
Santa Clara University

J. G. Linvill
Stanford University

H. Loomis
University of California

A. J. MacMillan
California State College

E. J. McCluskey
Stanford University

O. P. McDuff
University of Alabama

A. McKellar
Polytechnic Institute of Brooklyn

H. Mahrous
Pratt University

R. A. Manhart
University of Nevada

R. B. Marxheimer
San Francisco State College

R. H. Mattson
University of Arizona

Prof. Merz
Fairleigh Dickinson University

M. Miller
University of California/Berkeley

O. W. Muckenhirm
University of Toledo

J. Munushian
University of Southern California

R. H. Norris
Wichita State University

P. F. Ordung
University of California

A. E. Paige
University of Denver

S. R. Parker
U. S. Naval Post Graduate School

J. M. Pettit
Stanford University

W. H. Pierce
University of Louisville

J. L. Pokoski
University of New Hampshire

C. Polk
Stanford University

D. E. Rathbone
University of Idaho

W. L. Root
University of Michigan

D. Rummer
University of Kansas

J. C. Samuels
Howard University

J. E. Savage
Brown University

W. Schick
Fairleigh Dickinson University

R. H. Seacat
Texas Technological College

S. Seely
University of Massachusetts

C. B. Sharpe
University of Michigan

E. Sibert
Rice University

J. G. Skalnik
University of California

J. J. Skiles
University of Wisconsin

G. Sleman
University of Toronto

E. J. Smith
Polytechnic Institute of Brooklyn

L. D. Smullin
Massachusetts Institute of Technology

R. F. Soohoo
University of California

L. N. Stone
Oregon State University

W. H. Surber
Princeton University

A. K. Susskind
Lehigh University

W. F. Tatum
Southern Methodist University

M. O. Thurston
Ohio State University

J. T. Tou
University of Florida

F. Tuteur
Yale University

M. E. Van Valkenburg
Princeton University

R. Waid
University of Missouri

J. M. Walden
Oklahoma State University

J. W. Willhide
Boston University

Dr. R. Williams
University of Houston

F. Wood
Seattle University

L. A. Zadeh
International Business Machines Corp.

L. W. Zelby
University of Oklahoma