CHEMICAL ENGINEERING EDUCATION

CHEMICAL ENGINEERING DIVISION
THE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

March, 1963

CHEMICAL ENGINEERING EDUCATION

March 1963

Chemical Engineering Division
American Society for Engineering Education

CONTENTS

CHEMICAL ENGINEERING EDUCATION ®, Journal of the Chemical Engineering Division, American Society for Engineering Education. Published Quarterly , in March, June, September and December, by Albert H. Cooper, Editor.
Publication Office : University of Connecticut
P.O. Box 445, Storrs, Connecticut

Subscription Price, $2.00 per year.

COMPUTERS IN CHEMICAL ENGINEERING EDUCATION

P. B. Lederman (1), B. Carnahan, & G. B. Williams
Department of Chemical and Metallurgical Engineering
University of Michigan

The advent of computers has enabled the engineer to broaden his horizons with respect to the types of problems he may study and solve in a given amount of time. As an educational tool, the computer also proves very useful in that it takes over the routine calculations once an appropriate algorithm or procedure has been established for the problem at hand.

Students can investigate problems using numerical methods which before were not presented to them until they were able to solve them by compact analytical techniques. These methods are usually not presented in an undergraduate curriculum. One must, however, choose problems carefully so that the students avoid the numerous pitfalls present when numerical methods of solution are used.

Computers and Engineering Science

Considering the significant impact which the high speed electronic computer will have on our future technological development, several questions arise concerning the role of computers in engineering education. There seems little doubt that a good fraction of today's engineering graduates (engineers who may well be working as engineers in the year 2000) will have occasion to use computers in their technical work. Considering his probable imminent involvement with computers as part of his engineering work, the engineer must know a great deal more about computers than he can learn from the "giant brain" articles so prominent in the Sunday supplements. The question which comes to mind first is, Where should he learn about them? On the job or in the engineering school?

Those who feel that on-the-job training is adequate, usually claim that computer programming and computer-related work involves primarily techniques rather than engineering principle. Those who feel that the engineering school is a proper place for such training agree that there is a significant amount of technique (technique which will incidentally be useful in the student's future engineering work) but that the primary justification for such training, particularly at the undergraduate level, is based on the computer as an educational tool useful in the training of problem solvers. Viewed as an educational tool the computer can be considered a language for communicating as well as a machine for solving problems. A better understanding of principles can be attained because of the rigor required when communicating with the computer. This understanding is also reinforced because the student has a broader experience with solved problems.

Some features of a computer experience which seem to be related to the educational aspect of problem solving are:

a. Precise Definition: The computer is a rather rigid task master which requires precision in the statement of the problem and its method of solution. Preparation of procedures for computer solution introduces the student to a precise formal language (usually a mixture of English and algebraic notation). Because of the nature of such languages, the student's communication skill should be enhanced, he should tend to be more accurate, and he should achieve added understanding of mathematical notation and manipulation.

b. Logical Organization: Complex engineering problems require both an analytical ability (to subdivide the overall problem into simpler ones which can be handled) and an ability to synthesize (bring together solutions of individual parts as the solution of the whole). Preparing algorithms (problem solving procedures, flow diagrams) for a computer requires just such analysis and synthesis abilities.

c. Minimize Ambiguity: Because a computer solution requires the preparation of an orderly and detailed step-by-step procedure, the approach to the solution must be an unambiguous one (formal languages used by computers allow no ambiguity). No gaps in the logic are permitted.

---

(1) Present address: Esso Research Laboratories, Baton Rouge, Louisiana.

d. Recognition of Assumptions: During preparation of organized detailed procedures, assumptions which may be overlooked in a hand computation are frequently brought to the forefront. Of course, a bad assumption in a computer program has just as deleterious an effect as in a hand solution; however, because of the great computational speed, some assumptions necessary to permit hand computation may be removed entirely.

e. Solution of the General Problem: Because of the nature of the digital computer, i.e., its ability to read parameter values as data, it is usually possible (with little extra effort) to produce a general program which will solve a whole class of problems rather than a specific problem in a specific problem situation. This necessitates an essentially symbolic approach to problem solving and is rather different from the customary solution techniques involving mostly numbers. Such an approach requires a more abstract analysis which focuses on problem structure, rather than on "slide rule" details.

f. Problem Complexity: Because of high computational speed, the computer permits solution of significantly more complex (and hence, frequently more realistic) problems than can be "hand" solved. The drudgery of tedious repetitive calculations is removed. Unfortunately, it is usually wise (essential) to work at least one example problem in detail by hand for checkout purposes.

g. Numerical Solutions: The high speed computer solution permits numerical approximation of problems which are intractable analytically.

h. Logical Non-Numeric Problems: Since the digital computer is in fact a symbol manipulator rather than a mere number manipulator, it can solve a large class of logical, essentially non-numeric problems.

Computers and Engineers in Industry

Computers, both digital and analog, have wide acceptance in production, design and research. This trend, although still in its infancy, is making rapid strides. For example, the high degree of sophistication in some applications is illustrated in a recent announcement that complete engineering drawings fro roads are being turned out by computers. A recent survey for the American Petroleum Institute indicated that 86 of 127 responding refineries used off line computers, and several larger refineries have several computers working full time.

Today a large percentage of the "green light time" can be attributed to accounting and scheduling type functions in those computers associated with production units. More and more time is being used, however, by engineering groups to do repetitive computations and optimization studies. Several on line control computers are operating with some success. In a number of processes, where the reaction scheme is complex, for example, co-polymerization, there appears to be great incentive to use either open or closed loop computer control.

In research organizations computers are widely used in a number of areas. Although Esso Research Laboratories may not be typical because we are fortunate to have access to a great variety of computers, it is not atypical with respect to computer utilization. Therefore, we would like to take the liberty of using this organization as a basis for discussing the needs of the engineer vis-a-vis computers in industry.

A very brief look at the organization with particular emphasis on computers, as shown in Figurel, will help to orient the discussion. The Laboratory is one of the major development groups affiliated with Esso Research & Engineering Co. and does bench scale exploratory work as well as operate small and large pilot plants. To fulfill its mission it has several research groups, an engineering group and an applied mathematics group. Three digital computers are available, an IBM 1620 in the applied math group, an IBM 7074 in the Baton Rouge Refinery, and an IBM 7090 in Florham Park accessible by transceiver. The latter two installations are closed shop and the in-house facility is open shop. FORTRAN and symbolic assembly programs are available for all machines.

Almost all of the professional employees are involved with computers to some degree. Computers are used in two primary areas in our technical computing, data work-up and engineering studies. For the former some of the pilot plants are tied to a data gathering system. The data tape is used, along with data picked up from a daily analytical results tape, as input to any one of several unit data work-up programs. Much of the logic for these operations has been handled by experienced non-engineer programmers. The engineers must, however, supply algorithms for the unit work-ups. The programmers in the applied mathematics group are available to program the algorithms. It is usually more efficient, however, for the engineers to write and debug their own programs especially when, as is usually the case, program requirements change frequently.

The engineering studies include process optimization, reactor stability, control, reaction mechanism and similar studies which require the use of chemical engineering and related sciences together with a knowledge of mathematics and computer sense to bring them to a successful conclusion. This type of problem has increased in importance and will continue to take a greater share of the engineering talent and computer time in the future.

The demands on the engineer using computers in industry are many even though specialized help is usually close at hand. To maximize the information obtained from extensive pilot plant operations he must decide when computer data reduction with or without automatic data gathering is warranted, keeping in mind programming time required and computer cost. It is helpful if he can write and test his own programs because this cuts down lead time, often one of the greatest costs of research, and allows him to make any changes with the least delay.

This class of problems does not usually require extensive application of advanced mathematical techniques. It does demand a degree of rigor which we as chemical engineers were not able to exhibit before the computer era. Often the algorithms for this class of problems are simplified by use of simple matrix manipulation.

Engineering studies are becoming much more sophisticated and more encompassing in their scope. Engineers today should have a better understanding of the advanced mathematical techniques used to solve partial differential equations. In this area the computer is a great help, and at the same time its own worst enemy. Many engineers are not aware of the pitfalls which round off errors and non-convergence present. In addition to learning sound problem analysis and efficient algorithm construction, it is important that the engineer be made aware of the pitfalls involved in numerical methods and approximations so often used in digital computation.

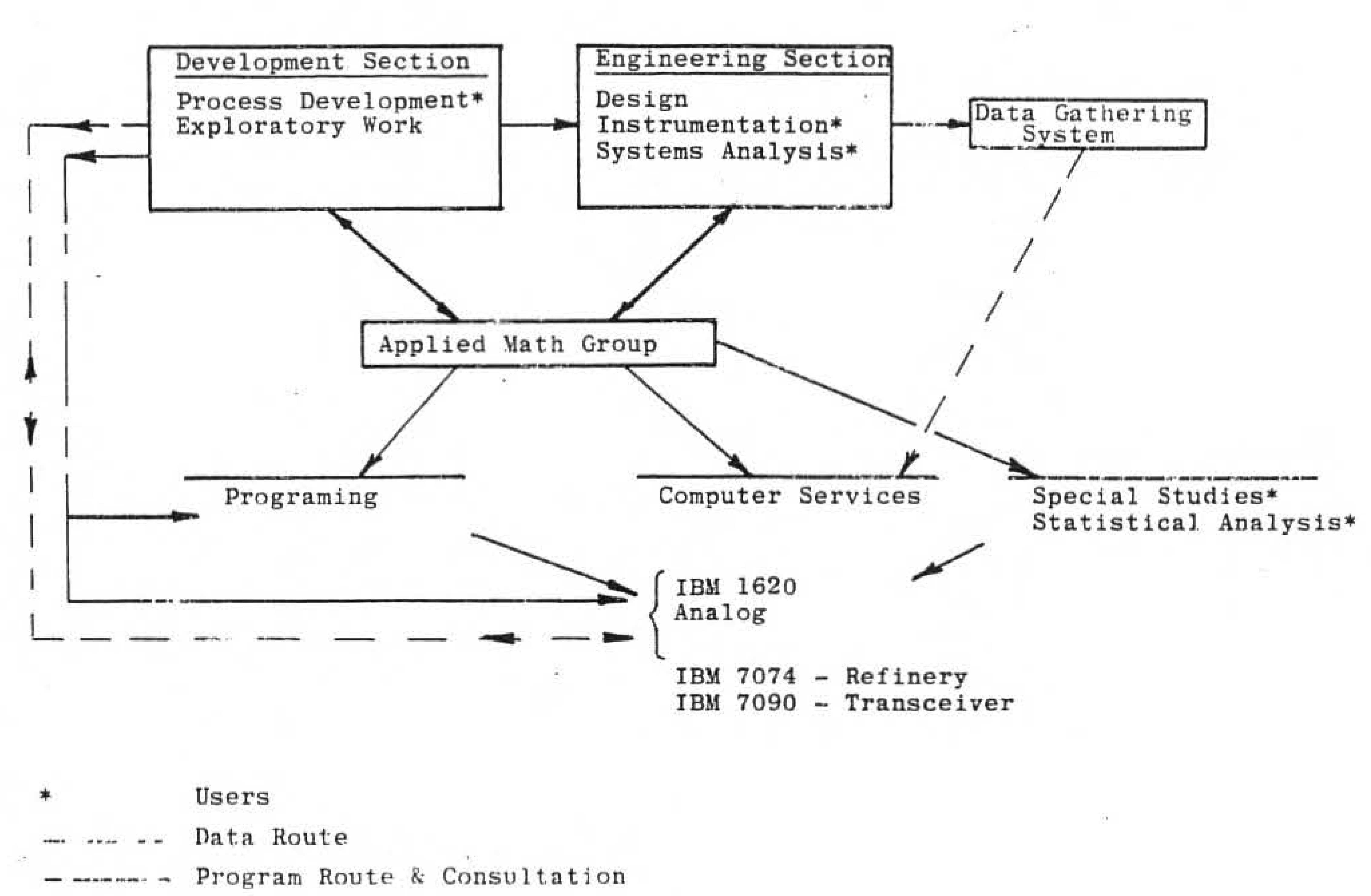
A third area where the chemical engineer and the computer have found common cause is in scheduling and economic optimization. This, of course, requires a knowledge of linear (or, in general, mathematical) programming. Today there is a need to develop a logic which can be used for ultimate design optimization by welding together and exercising supervisory control over independent routines representing a series of interdependent moduli or operations. Here again there is a need for a sound foundation in logic.

The use of computers by engineers in an industrial organization would not be complete without discussing communication between man and the computer. This has been very much simplified in the past few years at Michigan because of an excellent executive routine and a very versatile Algol language, MAD, with superb diagnostics. Due to the multiplicity of demands on most large industrial computers -- they do payrolls, accounting and complex engineering calculations -- and the limitations of the smaller computers such as the IBM 1620, communication is usually not quite as simple. This means that precompilation debugging should be more thorough. More important, it is most helpful if the engineer has some knowledge of computer operation or logic so that he can easily adapt to different computers, programming systems and methods of searching for errors.

The use of computers in engineering calculations and their introduction into the engineering sciences curricula is of great benefit to the young engineer. It forces him to be more analytical and rigorous in his approach to problems. It is important that along with the use of computers, numerical analysis, logic and some basic concepts of computer operation be introduced so that the engineer can make wise and efficient use of this powerful tool in an industrial atmosphere.

FIGURE 1

COMPUTERS IN AN INDUSTRIAL RESEARCH ORGANIZATION



Education With Computers

At Michigan there has been some contact with computers for the past 10 years. At first this was rather limited but since 1956 when an IBM 650 became available on an open-shop basis, the use of computers in the chemical engineering science curricula has been ever increasing. At first only limited use was made of the computer in graduate courses because access was rather difficult. With the arrival of a large computer, an IBM 704, and problem oriented languages such as FORTRAN and MAD, the computer became relatively accessible to undergraduates.

A question which arose is where and how the student should be introduced to the computer. If he is to gain a real computer proficiency, it appears that an introduction to computer organization and computer language should come early enough in his training to allow opportunity for extensive use of the machine in solving some engineering problems. Since it seems impractical (and probably unwise) to remove engineering course material to allow insertion of computer work into engineering courses, it would appear that the student should have an independent introductory course which gives him thorough training in the language and a general understanding of computing procedures. If he is not to be lost in the hopeless mire of detail there seems little doubt that the selected computer language should be of the problem-oriented rather than the machine-oriented type. If engineering classroom time is not to be wasted, then he must be trained well enough in the first course to eliminate the need for later retraining in the engineering classroom. The solution to this problem at The University of Michigan has been the introduction of a required one-hour course at the sophomore level which trains the student in the use of a problem-oriented language (MAD) and introduces him to some of the elementary numerical techniques.

At the present time at least one problem whose solution is best obtained with the aid of a high speed computer is presented in most every course in chemical engineering science. As can be seen in Figure 2 this means that students are exposed to computer methods from the beginning of their sophomore year. By applying the techniques learned in the course "Elementary Computer Techniques" immediately we find that students get a better appreciation of computer techniques. In each succeeding course one or more computer oriented problems are presented to the students. These problems, chosen by the individual instructor, are coordinated so that they illustrate many facets of computer programming and use.

Problems which arise in the assignment of computer problems as part of the engineering coursework homework load include timing. While there seems little doubt that a project-type assignment involving a time period of perhaps two weeks or more causes no significant difficulty, homework assignments done on a day-to-day basis do present some problems. Because of the nature of computer languages, i.e., the necessity for very precise grammar and punctuation, it is unusual for an undergraduate student to solve completely correctly an engineering problem on the first approach to the computer. The average may be three or four tries before success. The turn-around time, i.e., the elapsed time between submission of a program to the computer and its return for checking and possible resubmission in case of error, must consequently be fairly short if problems are to be completed between class meetings.

If computer integration into engineering classrooms is to be successful, the student's overall computer ability is certainly a major factor. The instructor's ability is probably even more important, particularly for undergraduate training. The selection of appropriate problems and the illustration (by example) of good computer habits (pointing out inadequacies or places where the computer should probably <u>not</u> be used, as well as where it should be) is essential.

In an attempt to help develop a better appreciation of the computer and computer techniques, a set of problems for use in an undergraduate chemical engineering curriculum is described in the Appendix. It is expected that the student will have had prior to or concurrently with the first of these problems a basic course in programming, as too much time taken from course content would be required to learn basic programming techniques.

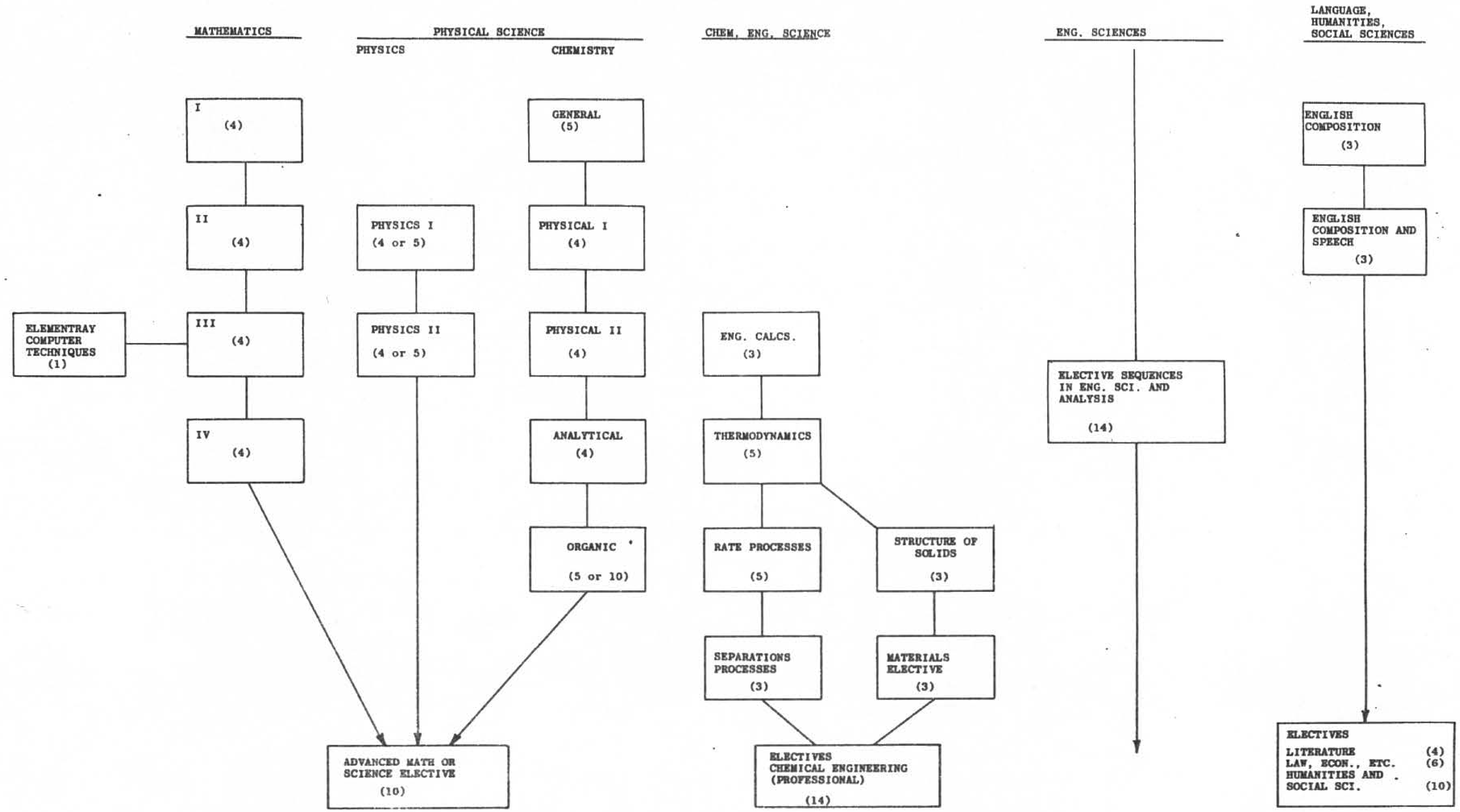
Conclusion

It is not wise to attempt to justify the use of a computer as a time saving device when one deals with a single problem operation, common to introductory educational endeavors. It is much better to look at the computer and program as a method of introcuding a tool which will enable the student, after some experience, to solve complex problems and will force the student into habits of careful, detailed problem analysis and logical solution methods. If exposure to computers and computer programming does nothing else it will be well worth the time and effort required if our students think more logically and precisely.

FIGURE 2

CHEMICAL ENGINEERING BSE PROGRAM

UNIVERSITY OF MICHIGAN



( ) SEMESTER HOURS REQUIRED.

BLOCKS ARE IN VERTICAL ORDER OF SEMESTERS IN WHICH COURSES ARE TAKEN.

It is true that this is a relatively new area in our curricula, but a very essential one. At this stage we have and should raise more questions than we can answer but by proper choice of problems the advantages of the computer will be well demonstrated. The student will gain new insight into many more problems and will become quite at home with the computer, a valuable tool for the engineer.

Acknowledgement

The authors appreciate the support of their colleagues in the Department of Chemical and Metallurgical Engineering at the University of Michigan and at Esso Research Laboratories for the many helpful discussions about their experiences. We are especially grateful to Professor D.L. Katz for his many helpful suggestions and Mr. E.A. McCracken for reviewing the original manuscript.

APPENDIX

A SERIES OF GRADED CHEMICAL ENGINEERING COMPUTER PROBLEMS

This series of computer problems is typical of those used during the past four years in the chemical engineering curricula at the University of Michigan. No attempt has been made to be all inclusive as this set, it is hoped, will merely serve as a guide. Complete solutions for many of these problems -- mainly written in MAD -- may be found in the various reports issued by the Ford Foundation Project on Computers in Engineering Education (2).

PROBLEM FOR A FIRST COURSE IN STOICHIOMETRY

A typical first problem after some basic programming experience might be a detailed mass balance as follows:

Problem Statement:

A countercurrent multiple-contact extraction system is to treat 100 tons per hour of tailings with fresh water as a solvent. The composition of the tailings fed to the extration unit is

Component	Mass Fraction
Water	0.48
Gangue	0.40
Salt	0.12

The strong solution leaving the system is to contain 0.15 mass fraction salt. A 99 per cent recovery of the salt is anticipated. Calculate the number of equilibrium stages required as a function of the solution retained by the gangue.

Solution:

This problem may be solved by a number of methods, including the method of linear differences. The basic material stage-wise balance method of solution is discussed in detail by Brown (1) and a computer solution for a similar problem may be found in the First Annual Report of the Project on the Use of Computers in Engineering Education sponsored by the Ford Foundation at the University of Michigan. (2)

Basically, the solution requires that an overall balance around the extraction be made and then stage-wise calculations be made until the raffinate from the last stage meets the required concentration specifications. Once the basic program is written, it may be easily enlarged to include either variable solution retention (as a function of the solution composition) or a series of solution retention values.

The problem is not difficult to program but does require the use of subscripts. Students have programmed a similar problem in 15 steps and with relatively little expenditure of time. It has the added advantage of illustrating the effect of solution carry-over.

A SECOND PROBLEM IN STOICHIOMETRY OR A FIRST PROBLEM IN THERMODYNAMICS

A second problem in Stoichiometry which could also serve as a first problem in Thermodynamics is the computation of the adiabatic flame temperature as a function of the fuel and fuel to air or oxygen ratio.

Solution:

The solution to this problem requires writing a material balance for the components involved and then making a thermal balance assuming a flame temperature. If the amount of sensible heat out equals heat of combustion plus the sensible heat in the correct flame temperature has been established. If the heat in and out do not balance, a new flame temperature must be chosen; the Newton-Raphson (3) method for estimating the successive values for the flame temperature will allow for quick convergence on the correct answer.

This problem is quite simple to program for a given fuel and fuel to air ratio and should require even less time than the problem discussed above. If it is desired to investigate the effect of fuel to air ratio or type of fuel, a little more thought is required to successfully program the problem. It is still a reasonable problem for people with a minimum of computer experience.

A PROBLEM IN THERMODYNAMICS

A number of thermodynamic calculations lend themselves readily to solution by the computer. The study of non-ideality and physical equilibria are probably two of the areas which are often neglected in beginning courses and which with the aid of the computer may be studied quantitatively. A typical request of students might be to require them to determine the degree of non-ideality and the per cent vaporization of a ternary mixture of hydrocarbons as a function of pressure and temperature.

Solution:

The instructor must supply an equation of state which is to be used to compute the properties of the non-ideal gases. This equation should be used to obtain densities and fugacities as a function of pressure and temperature. Once this has been programmed successfully, the students may use the equation of state in subroutine form.

The information, fugacity and density of the pure components and mixtures, obtained from the equation of state subroutine, may now be used in a larger program which actually calls on the subroutine when required. K's may be evaluated as a function of the fugacity and estimated composition and these may then be used in a flash-vaporization computation to calculate the degree of vaporization of a mixture. In addition gas phase densities may be checked against ideal densities to determine compressibilities factor values, a measure of non-ideality.

This program would be fairly advanced and it may be desirable to first compute the degree of vaporization based on Raoult's Law, that is, using vapor pressures obtained from a Clapyron relationship. This program would be quite simple to write and would serve as a good introduction to the general problems of physical equilibria.

A PROBLEM IN RATE PROCESSES

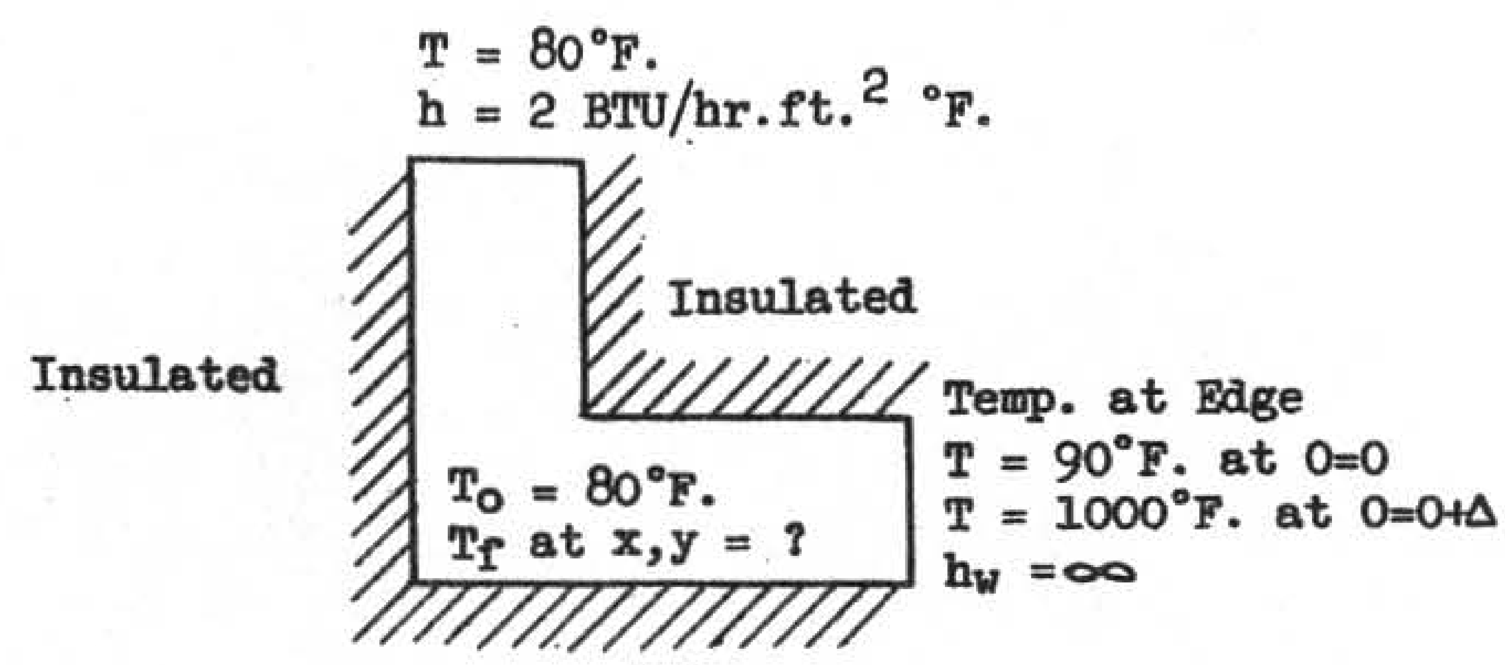
In the area of rate processes, the computer can greatly benefit and enhance the subject matter discussed in a beginning course. Many of the problems involving rates of heat or mass transfer involve partial differential equations which may be reduced to difference equations for numerical solution on the computer. A good elementary discussion of these methods may be found in "Numerical Methods for Science and Engineering" by Ralph G. Stanton.

The first problem in this area which is very suitable for computer solution is an unsteady-state heat transfer problem. The time required to quench an odd shaped bar is to be determined. In order to simplify the geometrical concepts, it is probably good to choose a rectangle with irregular insulation or study the converse problem of heating of a right angle bar at one end and determine the time dependence of the temperature at the other end of the angle iron as indicated in Figure 1 where T_f if a function of time.

This problem has been programmed by students quite successfully before. A detailed discussion of the method may be found in a paper by Rudd (4). In this problem a grid is established over the piece in question and the method of relaxation is employed to determine the temperature distribution over the grid at any given point in time.

FIGURE I

CROSS-SECTION OF RIGHT-ANGLE BAR



ANOTHER RATE PROCESSES PROBLEM

The design of tubular reactors is a very common task undertaken on a computer. In the case usually considered the equations representing the temperature and volume dependence of the system become somewhat complex for analytical solution. It is therefore necessary to use a set of difference equations and iterate down the length of the tube. This problem is well suited to computer solution but one word of caution. The choice of increment size is a difficult one. Care must be taken not to introduce round-off problems when results from preceding segments are used as a basis for computing the next segment.

A typical problem might be: the decomposition of SO_2Cl_2 to SO_2 and Cl_2. The heats of formation and heat capacities for the compounds are

SO_2Cl_2 (g) $\Delta H_f = -82{,}040$ cal/gm. mole

$$C_p = 13.00 + 24.0\times10^{-3}T - 14.4\times10^{-6}T^2$$

SO_2 (g) $\Delta H_f = -70{,}920$ cal/gm. mole

$$C_p = 8.12 + 6.825\times10^{-3}T - 2.103\times10^{-6}T^2$$

Cl_2 (g) $\Delta H_f = 0$

$$C_p = 7.5755 + 2.4244\times10^{-3}T - 0.965\times10^{-6}T^2$$

and the rate of decomposition may be expressed as follows:

$$r = A\, e^{E/RT}$$

where

$$A = 6.427\times10^{15} \quad 1/\text{sec}$$

$$E = -50610 \quad ^\circ\text{K cal/gm. mole}$$

Compute the length of a 1 1/2" I.D. tube required to insure 98% decomposition of 418 pounds per hour of SO_2Cl_2 fed at 200°F. and 1.2 atm, heat is transferred to the tube at the rate of 5000 BTU per square foot per hour.

DESIGN PROBLEM IN REACTION KINETICS

The use of computers in design courses has been quite successful. After a program of problems in earlier courses, the students often ask to have the computing facility made available to them. The choice of material taught or problems assigned to students is very wide so we include an example for illustrative purposes of what can be done.

A typical problem would be to determine the optimum reaction time for a given product if three competing products are formed, that is, mono-, di- and tri- ethanol amine. If one knows the rate constants as a function of temperature, one can write expressions for the concentration of all the species in the system. This results in this case in a system of five simultaneous linear differential equations. These may be readily solved using the standard Runga-Kutta method which is programmed for most large computing facilities and discussed in detail in any of a number of texts in numerical analysis (6).

The original reaction data for the ethanol amine reactions were determined by Ferrero and coworkers (7) and are summarized below with additional data required to solve the problem. The possible reactions are:

$$\mathrm{O}\begin{array}{l}\diagup \mathrm{CH_2}\\ \,|\\ \diagdown \mathrm{CH_2}\end{array} + \mathrm{NH_3} \xrightarrow{k_1} \mathrm{NH_2CH_2CH_2OH}$$

$$\mathrm{O}\begin{array}{l}\diagup \mathrm{CH_2}\\ \,|\\ \diagdown \mathrm{CH_2}\end{array} + \mathrm{NH_2CH_2OH} \xrightarrow{k_2} \mathrm{NH(CH_2CH_2OH)_2}$$

$$\mathrm{O}\begin{array}{l}\diagup \mathrm{CH_2}\\ \,|\\ \diagdown \mathrm{CH_2}\end{array} + \mathrm{NH(CH_2CH_2OH)_2} \xrightarrow{k_3} \mathrm{N(CH_2CH_2OH)_3}$$

$$\mathrm{N(CH_2CH_2OH)_3} \xrightarrow{k_4} \text{higher products}$$

The velocity constants for the Arrhenius equation

$$k_i = A_i e^{-E_i/RT} \quad \text{are:}$$

	A gm. mole/liter min.	E
1.	3.58×10^8	14,500
2.	9.9×10^9	15,600
3.	2.58×10^9	15,000
4.	3.93×10^6	12,650

It is now possible to determine concentrations of each product in the effluent of an isothermal reactor as a function of the space velocity and the initial concentrations. If sufficient data is available on the heats of reaction and specific heats of the constituents the non-isothermal case may be studied.

This problem and similar problems have been very useful in demonstrating the course of reactions. This and similar problems, it should be mentioned, are also very suitable for analog computer analysis.

REFERENCES

1. Brown, G.G., et.al., Unit Operations, John Wiley, New York, 1950.

2. Katz, D.L. and Organick, E.I. (ed.), Electronic Computers in Engineering Education, First Annual Report to The Ford Foundation, Ann Arbor, 1960.

3. Stanton, R.G., Numerical Methods for Science and Engineering, Prentice-Hall, New York, 1961.

4. Rudd, D.F., "Temperature Distribution in a Conducting Solid," in First Annual Report on Computers, Ford Foundation Project, Ann Arbor, 1960.

5. Burnett, G.M., Mechanism of Polymer Reactions, Interscience, New York, 1954.

6. Ralston, A. and H.S. Wiff (ed.), Mathematical Methods for Digital Computers.

7. Ferrero, P., F. Berbe and L.F.Flamme, Bul. Soc. Chim. Belg. 56, 349-368 (1947).

8. "Use of Computers in Engineering Education," Second Annual Report, Ford Foundation Project, Univ. of Michigan, Dec. 15, 1961.

The following list of problems were prepared by various staff members and visiting professors at the University of Michigan under the sponsorship of the Ford Foundation project on "The Use of Computers in Engineering Education" (8). The titles are included to suggest some more areas where computers have been and can be used. Complete descriptions and programs are available from the project at the University of Michigan.

Title	Level[1]
Optimization of Reactor Operation	2
Approach to Steady-State of an Othmer Still	2
Temperatures and Heat Flux in a Radiant Thermal Circuit	2
Heat Balance for an Iron Blast Furnace	2
Three Component, Two Phase, Counter-Current, Liquid Extraction	3
Temperature Distribution in a Three Dimensional Body	3
Solution of a Boundary Value Porblem Using an Initial Value Technique: Temperature Profile in a Circular Transverse Fin	3
Velocity Profiles for Flow in Smooth Pipes	3
Determine Reflux Ratio by McCabe-Thiele Method	3
Temperature Profile in a Longitudinal Fin Using the Analog Computer	3
Diffusion and Slow Chemical Reaction	4
Number of Theoretical Plates in a Multi-Component Distillation Column	4
Multiple Regression Analysis	4
Solvent Allocation in Multi-Stage Crosscurrent Extraction	4-G
Dynamic Heat Exchange	3-4
Storage of Natural Gas in Aquifers	G
Adiabatic Reactor	G
Predicting the Scrap Requirement for the Oxygen-Steel Converting Process	4

(1) Year in which normally used.

The Integrated Use of the Digital Computer in Chemical Engineering Education

by

Paul T. Shannon
Purdue University

Introduction

"How do you teach a robot to perform process design and optimization calculations involving recycle streams for any arbitrary process sequence for an arbitrary set of equipment?" Our robot has many desirable qualities as well as limitations. He possesses the ability to do simple arithmetic extremely fast and can remember everything that he is told. He will answer simple questions which have been unambiguously encoded to him in a form that they are either "yes" or "no" and he will do exactly as he is told to do. This last quality is both an asset and a drawback, as any who have done computer programming will testify.

Our robot is not yet built to "see" but he may have this ability in the not too distant future. At present, however, he will deal only in numbers. Thus, if we wish to have our robot use any of the visual aids which we as engineers have found so useful in the study of complex engineering problems, we must find a suitable method for numerically encoding our graphical techniques for communication to and use by our robot. We will find that the answer to this problem, of numerically encoding graphical aids and their subsequent uses is the key to our problem of robot education in the field of systems analysis.

Digital computer programming has evolved as a criterion of excellence of understanding. If you are able to tell an idiot how to perform a given calculation, taking into account all the possibilities and ramifications of the problem, then you, as the programmer, truly understand the calculation yourself. I think no one will argue that one of the most effective ways to teach a student a given calculational procedure is to ask him to program the problem on a digital computer and assist him in doing so.

Digital computer programming is a tremendous amount of work. Even simple problems require a good deal of time and effort and those who have not actually programmed several problems and experienced the frustrations and elations of debugging have the pat answer, "We'll just solve that set of equations on a computer." A digital computer program works or it doesn't work. One receives almost no partial credit.

It has been argued that the student should write each of the digital computer programs he is to use. Timewise this is just not possible. The engineer in practice or the student in school must of necessity use programs written by others. This, also, is a considerable amount of work but not nearly that involved in writing the program itself.

If the digital computer is to be used extensively by the undergraduate student, he must be shown that it is a tool of significant help to him in his course work. To illustrate this point, the successful use of the digital computer in the Chemical Engineering Laboratory courses at Purdue will be described. Next we shall consider our question of robot education. A description will be given of the computer executive system currently under development at Purdue aimed at answering the posed question. This executive system allows the arbitrary sequencing of digital computer programs and thus enables the user to write the computer program for a given problem in essentially the time required in formulating the problem to be solved. We will find that we arrive at some very powerful generalizations and some fundamental concepts which have bearing on the teaching of chemical engineering. It will be shown that the executive system develops a basic conceptual and calculational frame work which is easily taught to undergraduate students.

Use in the Chemical Engineering Laboratory

Digital computer programming is presently taught and very effectively used in the Chemical Engineering laboratory course at Purdue. For most students this is their first introduction to programming since there is not a required course in programming and numerical methods for all engineering students in their freshman or sophomore year. In Chemical Engineering, an elective one credit hour course in computer programming is offered in the fall semester. This course offers the students additional computer experience and has been taken by about 70 students, both undergraduate and graduate, each fall.

The first 2-2 1/2 weeks of the Chemical Engineering Laboratory course is spent on an introduction to digital computer programming. Each student is required to write and run one or two simple programs. Then the "canned" computer programs (developed by the students and the staff during the last three years) which will be used in conjunction with the laboratory experiments are explained in detail. The laboratory time spent on computer programming is "made up" in that the time formerly allowed during the course for experimental calculations and report writing is correspondingly decreased. In fact, the students now do an additional experiment and perform more runs in a given experiment than before. Sample results from the computer for three of the programs are shown in Appendix A.

Perhaps of greatest significance have been the very good "side effects" accompanying the integrated use of computers in the laboratory. First, the majority of what the students referred to as the "mickeymouse" and "dog labor" has been eliminated with a corresponding significant increase in student interest and enthusiasm for the course. Second, each computer program has been written incorporating an error analysis. This forces the student to consider the accuracy of his experimental measurements and their effect on his calculated results. Many a student has been amazed when his "simple" heat balances on an exchanger turn out to have an error 50-100%. Third, only the input data need be checked in order to check the complete calculations thus eliminating "fudged" results. Finally, since about 90% of the required calculations are done by the "high speed idiot" primary attention can be focused on the more important questions of "what do you want to measure and how accurately can you measure it?" and "what is the significance and use of your results?"

There are, of course, problems associated with the use of "canned" programs in the laboratory. One is always concerned that the students understand how the calculations are performed and does not merely follow a "work book" procedure dictated by write-up of the computer program. The laboratory equipment and the large classes, requiring three or four laboratory sections each semester, had made the course quite formalized even before the computer programs were used. It was these facts that justified the effort of writing the computer programs in the first place. Realizing this, the staff has been introducing variety into the laboratory not by constantly changing the equipment and asking each group to perform a different experiment, but by asking the students a variety of questions regarding the interpretation and use of their experimental results.

In summary, the integrated use of the digital computer in the Chemical Engineering Laboratory has been very successful. The previously boring, repetitious calculational portion of the course has been eliminated, and the students first encounter with a digital computer is its use as a logical means to an end rather than an end in itself.

The subsequent use of the digital computer by the students in other chemical engineering courses at Purdue has been very limited. Let us turn out attention to the generalized executive computer program for doing process design calculations which is currently being developed at Purdue. Following a detailed presentation of executive program, it will be shown how the program serves as a natural guide for the truly integrated use of the digital computer in the undergraduate curriculum.

The PACER Executive System

PACER - Process Assembly Case Evaluator Routine

- is a digital computer executive program. PACER is being written to include eventually three major phases: (1) material and energy balances - transient and/or steady state; (2) economic analysis; and (3) internal parameter variation. The PACER program is primarily one which furnishes communication between the equipment subroutines, does a tremendous amount of internal bookkeeping and has the ability to do trial and error recycle calculations automatically. It is similar in its function and purpose to engineering calculational programs currently being developed and used by companies such as C. F. Braun, Humble Oil and Refining, M. W. Kellogg, Phillips Petroleum, Shell Oil and Union Carbide. Many of the basic ideas employed in PACER came from a paper presented by M. T. Tayyabkham of Union Carbide Corporation entitled "Simulating Unsteady State Operation of a plant on a Digital Computer" which was presented at the AIChE meeting in Cleveland, Ohio, on May 9, 1961.

The PACER program was developed by studying the structure of the problem of performing process calculations so that we could teach our robot to do the calculations for us in the future. The system design specifications which evolved are shown in Table 1. The major features of PACER are in the use of Stream and Equipment matrices for handling all the information associated with a problem, modular programming, and the use of the Process Matrix by which the processing sequence itself is supplied as input data. Table 2 gives the definition of these and other closely related terms.

Table 1

System Design Specifications
for
Process Assembly Case Evaluator Routine

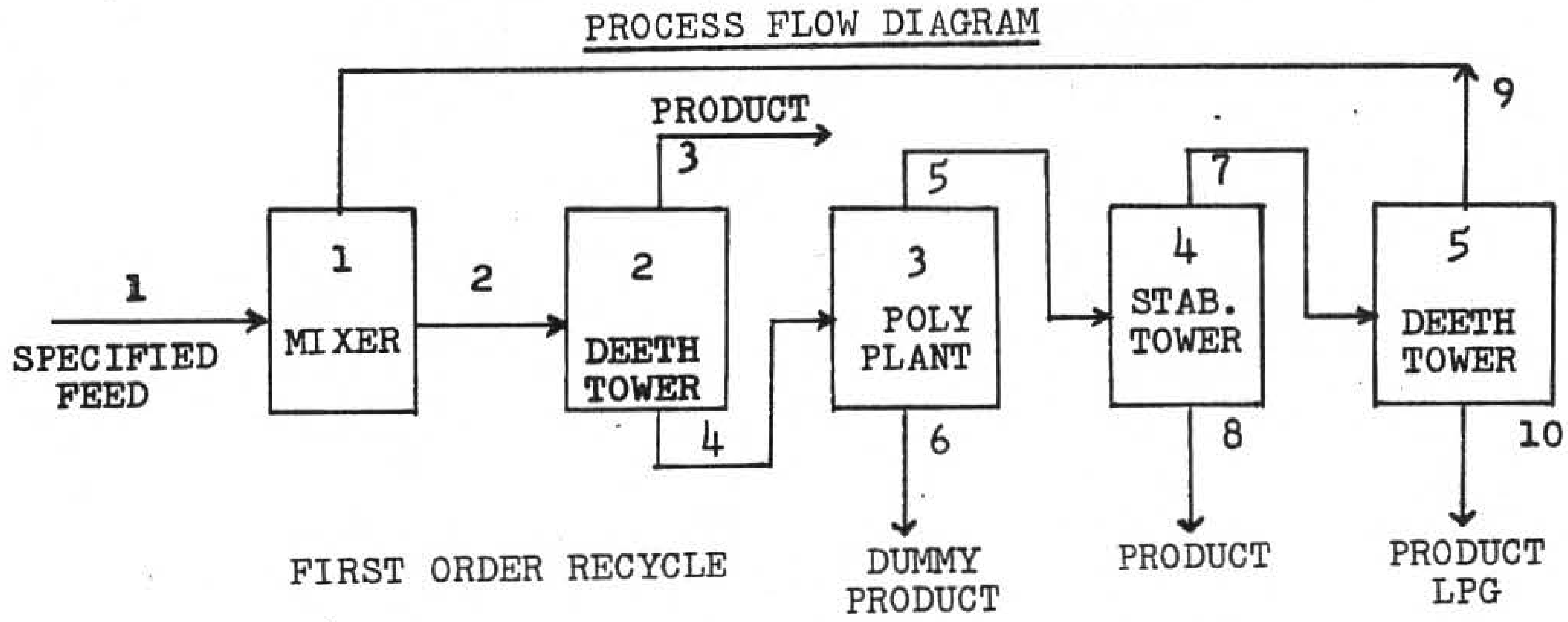
1. Process Sequence, Set of Equipment, Boundary Conditions to be supplied as Input Data
2. Process Sequence, Equipment Used and /or Operating Conditions to be easily changed by user
3. PACER is to determine calculations required, do them, and print requested results
4. PACER is to be able to do both steady-state and transient behavier calculations
5. PACER must be open ended, i.e., able to be expanded and modified as required
6. PACER must be useable during development
7. PACER must be able to incorporate all past work easily
8. PACER must be easy to use and understand
9. PACER ultimately must have very large information retrieval capacity

The Process Matrix is the numerically encoded process flow diagram. It is the heart of the PACER system. By its use, the processing sequence becomes part of the input data. For any given process such as shown in Figure 1, the Process Matrix is readily formed as follows. First each piece of equipment is numbered. Then each stream on the diagram is numbered. The numbering of the equipment is arbitrary except for one or two restrictions when complex second and third order branched parallel equipment loops are involved. The numbering of streams is also arbitrary. However, particular equipment programs may require a certain ordering of stream numbers in the Process Matrix such as noted in Figure 1. These are very minor restrictions and do not limit the PACER program. The equipment numbers are given in the first column of the matrix. Then for each equipment number, the associated input streams numbers (as positive numbers) followed by the output streams numbers (as negative numbers) are listed across the row. The Process Matrix for the flow diagram is also shown in Figure 1. A second more complex example is shown in Figure 2.

FIGURE 1

PROCESS SIMULATION - SAMPLE PROBLEM #1

PROCESS FLOW DIAGRAM



PROCESS MATRIX

No.	Equipment Name (Subroutine)	Associated Streams		
1	MIXER (UNAME3)	1	9	-2
2	TOWER (UNAME2)	2	-3	-4
3	REACTORS (UNAME2)	4	-5	-6
4	TOWER (UNAME2)	5	-7	-8
5	TOWER (UNAME2)	7	-9	-10

NOTE: In subroutine UNAME2:
First output stream number is OVERHEADS.
Second output stream number is BOTTOMS.

Table 2. Definition of Terms

STREAM $\equiv$ ANY CHANNEL OF INFORMATION

The information is contained in the following matrices.

STREAM PARAMETERS LIST

List of values of those variables specifying what you are talking about.

STREAM CONTROL VARIABLES LIST

A list of values of those variables associated with "controlling" the information in the parameters list.

STREAM PARAMETERS MATRIX SN

The composite list of all stream parameter lists. Stream number = matrix row number.

STREAM CONTROL VARIABLES MATRIX SNC

The composite list of all stream control variables lists. Stream number = matrix row number.

STREAM FLAG LIST

A list of numbers, one for each stream, which signals its status. Stream number = list row number.

NOTE: Format of SN and SNC should be the same for all streams to allow interchange between equipment.

Stream formats usually determined by equipment subroutines.

The decision of which variables are "parameters" and which are "control" is arbitrary.

EQUIPMENT — ANY MATHEMATICAL MODEL OR SET OF RULES
— AN INFORMATION MODIFIER
— A UNIT CALCULATION

An EQUIPMENT takes known input information contained in input streams and uses mathematical rules to produce output information in output streams

EQUIPMENT PARAMETERS LIST

List of values of variables associated with specifying basic "size" and mode of operation of the EQUIPMENT

EQUIPMENT CONTROL VARIABLES LIST

List of values of variables controlling equipment operation.

EQUIPMENT PARAMETERS MATRIX EN

Composite list of all equipment parameters lists. Equipment number = matrix row number.

EQUIPMENT CONTROL VARIABLES MATRIX ENC

Composite list of all equipment control variables lists. Equipment number = matrix row number.

EQUIPMENT FLAG LIST

List of numbers, one for each equipment, which signals its status. Equipment number = list row number.

Note: Format of Equipment Parameters and Control Variables Lists need not be the same for all equipment.

Equipment always written as a subroutine.

The decision of which variables are parameters and which are control variables is arbitrary.

PROCESS MATRIX $\equiv$ NUMERICALLY ENCODED PROCESS FLOW DIAGRAM

An array of the Equipment numbers and their associated input (positive) and output (negative) stream numbers.

The Process Matrix defines the System (Process sequence) for PACER and is used to determine how the calculations will be done.

The calculational procedure is independent of:

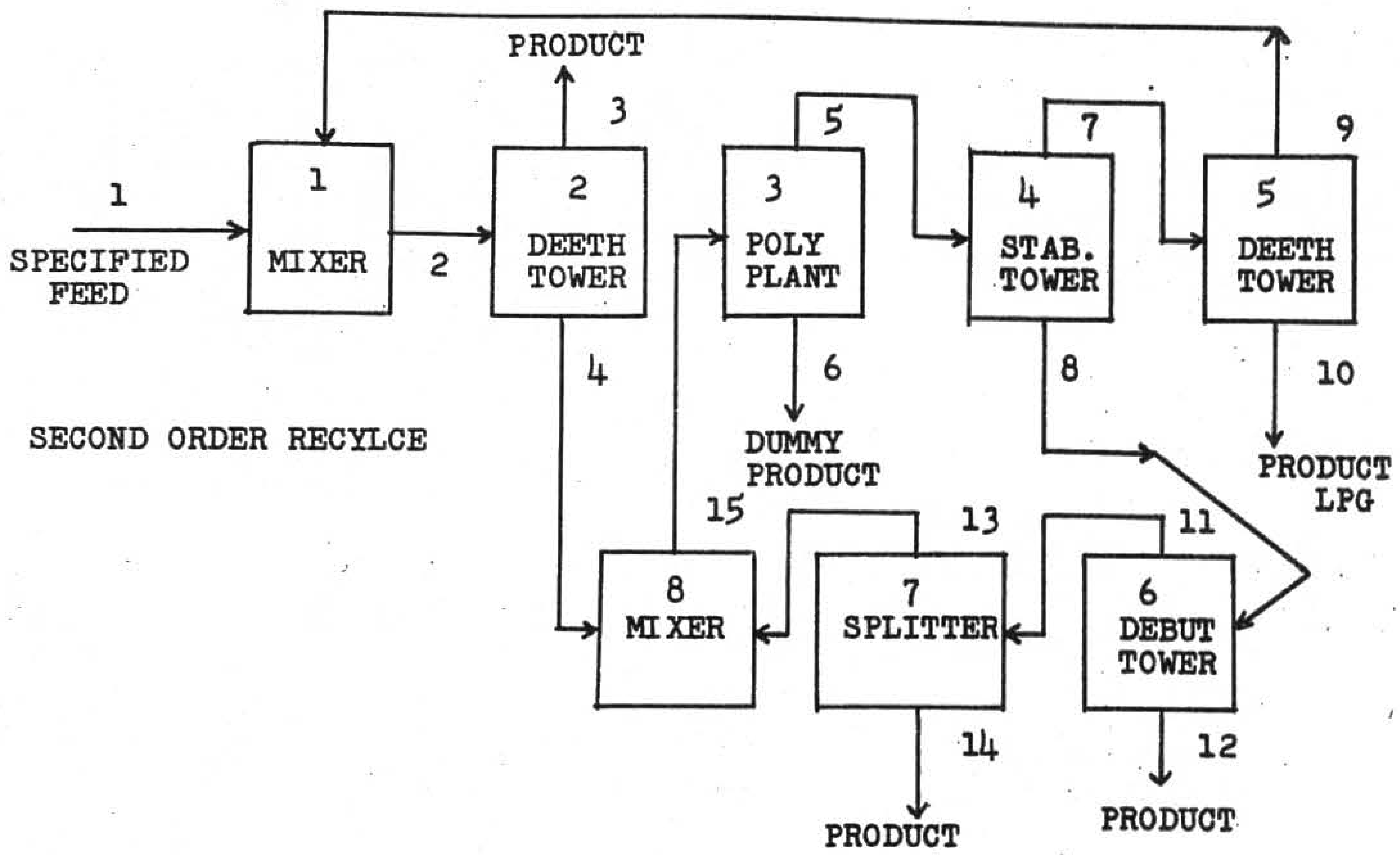
1. the sequence of equipment in Process Matrix

2. the numbering of the streams and equipment

FIGURE 2

PROCESS SIMULATION - SAMPLE PROBLEM #2

PROCESS FLOW DIAGRAM



PROCESS MATRIX

Equipment No.	Name(Subroutine)	Associated Streams		
1	MIXER(UNAME3)	1	9	-2
2	TOWER*(UNAME2)	2	-3	-4
3	REACTORS(UNAME2)	15	-5	-6
4	TOWER(UNAME2)	5	-7	-8
5	TOWER(UNAME2)	7	-9	-10
6	TOWER(UNAME2)	8	-11	-12
7	SPLITTER(UNAME2)	11	-13	-14
8	MIXER(UNAME3)	4	13	-15

*Note: 1st Output Stream Number is Overheads
2nd Output Stream Number is Bottoms

Table 3. Input Data to PACER

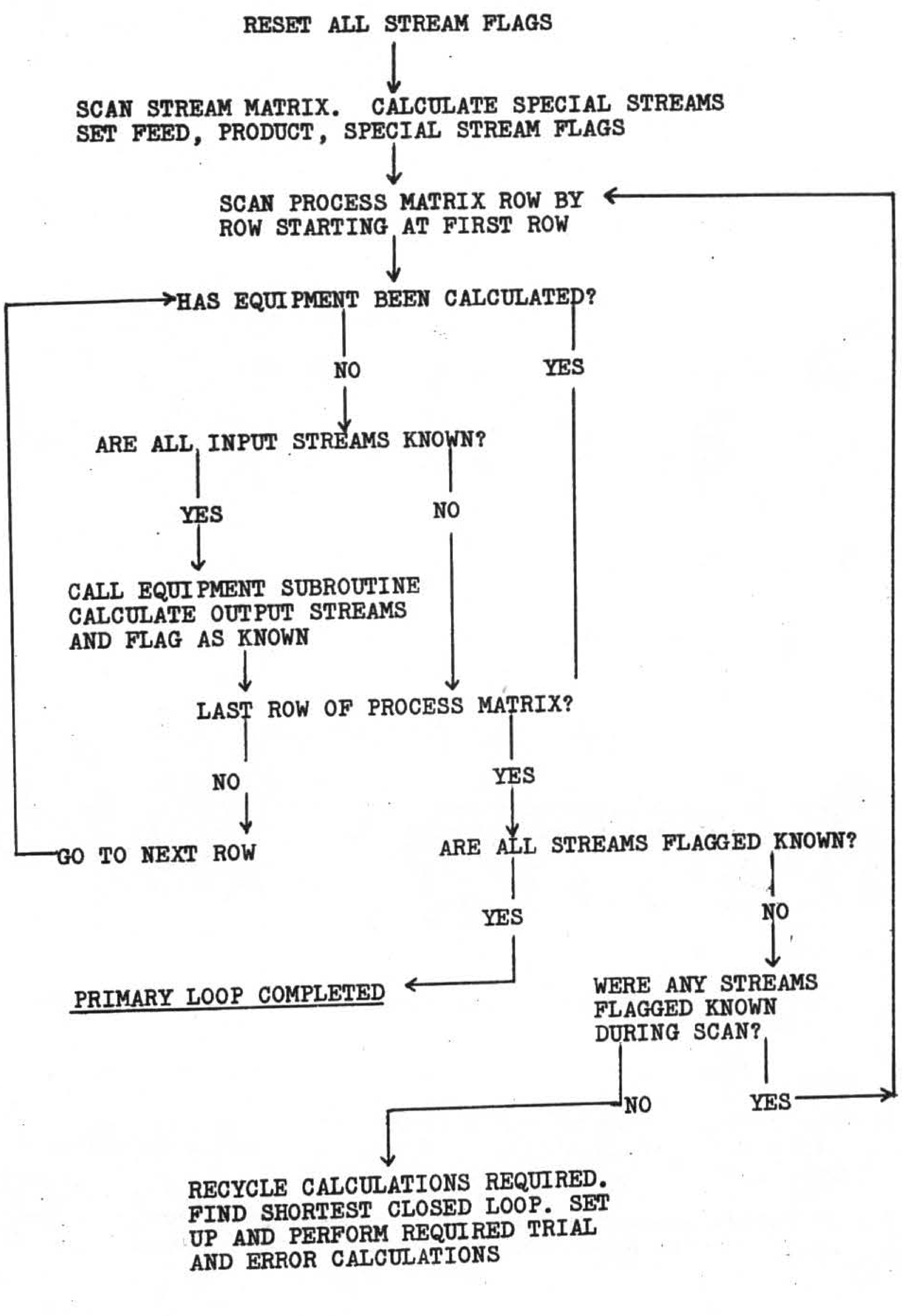
1. Control Cards describing the size of the problem, number cases to be evaluated and print out desired
2. The Process Matrix. One card for each equipment giving the process matrix row number, the equipment number, name of corresponding subroutine and the associated stream numbers
3. The Equipment Parameter and Equipment Control Variables Matrices
4. The Stream Parameter and Stream Control Variables Matrices
5. The Stream Parameter and Stream Control Variables Test Lists used for testing for convergence in the trial and error recycle computations
6. Preferred Stream Numbers, if any, used in determining the calculational procedure for recycle computations.

The PACER calculational procedure is independent of the numbering of the equipment and streams. It is also independent of the order of the equipment in the Process Matrix. The basic program logic of the PACER system is shown in Figures 3 and 4. Space and time prohibit a detailed description of PACER which has been written in FORTRAN for an IBM 7090 computer and consists of well over 1000 FORTRAN statements. The description and discussion presented here are independent of the programming language and machine.

As shown in Figures 3 and 4, the Process Matrix is used to determine automatically the sequence of Equipment calculations thus providing the logical relationships for computations. The basic calculational rule used is that if the input streams are known or have been computed, then the Equipment subroutine can calculate the corresponding output streams. The streams are treated as entities in that if a stream is known, all variables in the corresponding Stream Parameter and Stream Control Variables lists are taken as known, and, when a stream is calculated, the change in any and all variables are computed by the particular unit calculation alone. During each primary loop the Stream Flags and Equipment Flags are used to indicate the calculational status of each stream and equipment. PACER uses these flags to make decisions in determining the calculational procedure. Following the logic shown in Figures 3 and 4, one would discover that Figure 1 is a first order recycle problem where stream 9 must be assumed known in order to do

FIGURE 3

PROGRAM LOGIC FOR PRIMARY LOOP



the calculations. For Sample Problem #2 in Figure 2, Streams 9 and 15 must be assumed known so that it is a second order recycle problem.[1]

The values given in the Equipment and Stream Matrices represent a snapshot picture of the entire process at any given time. In each primary loop, the values of the variables are recomputed. For time independent problems, the primary loop is calculated only once. In transient behavior problems, each primary loop corresponds to a small interval of time. The user supplies the initial boundary conditions and corresponding steady state answers are obtained by repeated primary loop calculations. Tayyabkham discusses the unsteady state case in detail. In his paper he cites the simulation of an organic chemical plant consisting of 50 flow streams and 30 pieces of equipment on a IBM 7090 computer. About 100 subroutines were used. Each primary loop, equivalent to 0.1 hours, could be calculated about 20 to 25 times a minute so that 2 hours of computer time corresponds to about 300 hours of real time.

As seen in Figures 3 and 4, the Process Matrix is scanned several times in completing a primary loop. PACER recognizes the need for trial and error recycle calculations when no additional equipment can be calculated by repeated scanning of the Process Matrix and all streams have not been calculated. The executive program then makes an ordered list of unknown input streams. The list is ordered in the sense that the user can supply PACER with a list of preferred stream numbers and these stream numbers are placed at the top of the unknown stream list. PACER then takes the first stream number in the unknown stream list, flags the stream as temporarily known and scans the Process Matrix to see if an equipment calculational loop can be found such that the assumed known stream would be calculated as an output stream. If this is not possible, then the next stream in the unknown stream list is assumed known and the search repeated. First all single streams are assumed known one at a time in trying to find a calculational path. Then all possible combinations of two streams are tried and finally all possible combinations of three streams. The order of the recycle equals the number of streams which have to be assumed in order to establish a closed calculational loop.

Once a calculational loop consisting of an ordered list of equipment numbers is found such that all assumed known streams are calculated output streams, PACER eliminates all equipment from the list which is not required to complete the calculational loop. This is done since non-required equipment merely increases calculational time. The present values of the variables are taken as the starting values for the trial and error recycle calculations. Once the shortest order list of equipment has been determined, the necessary booking is done and the equipment repeatedly calculated in order until convergence is obtained. Following some additional bookkeeping, PACER returns to the primary loop scan to complete the calculations for the primary loop.

It is emphasized that the term Equipment and Streams must be used in the broadest sense of their definition in order to see the scope and magnitude of the PACER system. An Equipment subroutine may or may not correspond to a piece of real hardware such as a heat exchanger or distillation tower. It can be any set of mathematical rules. For example, it might be a program for changing the format of Stream Parameters and Control Variables Lists so as to be able to incorporate a previously written subroutine into the PACER library without having to rewrite the subroutine. In the same manner, the definition of Stream is used in the broadest sense.

The requirement of a uniform format for all Stream Parameter and Stream Control Variables lists allows the complete interchange of Streams and Equipments. This gives the PACER user the ability to quickly study various process sequences and so be able to handle a whole new class of problems on the computer. The PACER executive program is independent of the Stream and Equipment Matrices format in that it will handle any consistent set.

Appendix B gives a brief description of two simple Equipment subroutines and the format of their associated Stream and Equipment matrices. The relationship between the equipment subroutines and the formats of the Stream and Equipment matrices is typical. These two Equipment and subroutines are sufficient to simulate the processes shown in Figures 1 and 2. These two simple programs and PACER, could be used effectively to illustrate many, if not most, of the aspects of process design and optimization for students in their first course in chemical engineering.

---

[1] Mr. Henry Mosler, one of my graduate students at Purdue who is developing PACER, pointed out that Sample Problem 2 is only a first order recycle problem. This was "discovered" when PACER solved the problem. Streams 5 and 15 are common to the two calculational loops. Assuming 5 or 15 as known, all equipment can be calculated. The author, and perhaps the reader, had been conditioned not to guess intermediate streams in this type of problem.

One of the powerful features of the PACER system lies in the fact that all the equipment calculations are written as subroutines. Each equipment subroutine is a "unit calculation", a simple extension of the fundamental chemical engineering concept of the "unit operation." This gives rise to the concept of modularity in programming which is extremely useful. The subroutines are "complete" in the sense that they are written to calculate the output streams using only the data contained in the input streams and the Equipment Parameters and Control Variables lests. Hence a library of equipment unit calculations may be developed consistent with a basic stream and equipment matrices format. Several unit calculations of increasing complexity can be written for a given type of equipment such as a distillation tower calculation. Then the user of PACER may choose the level of mathematical description required by his particular problem. Another great advantage lies in the fact that only one subroutine is required for a given equipment no matter how many times that equipment is used in a process.

Input information for the PACER executive program consists of the data listed in Table 3. The routine for reading the input data is such that individual rows in the Process, Equipment, and Stream Matrices are read in separately. Thus, the user can easily change any portion of these matrices with a minimum of effort. If additional equipment subroutines not in the PACER library are to be used, these are read into the computer prior to the data called for in Table 3. The Process Matrix specifies the problem to be solved and is used to determine the method of solution. The Stream and Equipment Matrices supply the initial boundary conditions for the case to be evaluated and after solution, contain the computed results.

With PACER almost all the "bookkeeping" is done within the computer. The program user does not have to be directly concerned with all the intermediate stream results. Previously, assuming the equipment subroutines were available in "closed form," the user would have to prepare the data sheets for the first program, punch the cards, run the problem, examine the results, prepare the data cards for the second program (which probably require a different format), run the problem, etc. This took all of his time and effort. He would only have time to make one or two complete process calculations; his calculations would not have

FIGURE 4

PRIMARY LOGIC FOR RECYCLE CALCULATIONS

FORM ORDER LIST OF UNKNOWN INPUT STREAMS
↓
CHOOSE INPUT STREAMS ASSUMED "KNOWN"
TRY ALL COMBINATIONS OF 1,2 or 3 STREAMS
RECYCLE ORDER = NUMBER OF STREAMS ASSUMED
↓
SCAN PROCESS MATRIX. CAN ASSUMED INPUT STREAMS BE CALCULATED AS OUTPUT STREAMS?

NO ↓ MODIFY CHOICE OF "KNOWN" INPUT STREAMS

YES ↓ DETERMINE SHORTEST, ORDERED LIST OF EQUIPMENT HAVING ASSUMED INPUT STREAMS AS OUTPUT STREAMS
↓
DO BOOKKEPING REQUIRED FOR ITERATIVE RECYCLE CALCULATION
↓
SELECT EQUIPMENT FROM ORDERED LIST
↓
CALL EQUIPMENT SUBROUTINE AND CALCULATE OUTPUT STREAMS
↓
TEST OUTPUT STREAMS FOR CONVERGENCE
↓
NO — HAS ALL EQUIPMENT BEEN CALCULATED? — YES
↓
YES — HAVE ALL STREAMS CONVERGED? — NO
↓
TRANSFER RESULTS TO STREAM MATRIX
↓
RETURN TO PRIMARY LOOP

converged, and he would then "adjust" his answers and report his results. This need no longer be the case.

When the user of PACER specifies the Process Matrix he has written a complete digital computer program in the time it took him to write the Process Matrix. In most cases, the resulting computer program is of such magnitude as to have been prohibitive if it had had to be written from the beginning. The great advantage of modularity and the "open ended" approach in programming as used in PACER is illustrated in the fact that for PACER, Sample Problems #1 and #2 are the "same" problem. It is the same problem for any specification of the unit calculation equipment subroutines to be used. Thus PACER' and a library of equipment subroutines gives the user a tremendous computational ability.

While it has not been possible to present a detailed description of the PACER system it is hoped that a clear picture has been given of the basic approach and fundamental concepts involved. It is hoped to have the first phase of PACER operational on the IBM 7044 at Purdue next Spring. A more detailed description of the PACER system will be made available at that time.

Now let us consider how the PACER system might be effectively used in the undergraduate chemical engineering curriculum. At the same time we shall see how this would lead to the integrated use of digital computers in the students undergraduate courses and be of tremendous use in his senior year courses in Process Design and Process Dynamics and Control.

The basic ideas and operation of PACER are simple and could easily be taught to students in their first course in chemical engineering. What more effective way to teach a student process recycle calculations then have him learn to teach a stupid robot? The use of the Stream and Equipment Matrices certainly represents an orderly way to handle the information associated with a problem. Using simple unit calculations such as given in Appendix B, the students could very quickly be introduced to the concepts and problems of process design. The decision aspects of engineering and the description function of science are dramatically illustrated in this approach. The Equipment Unit calculations are the scientific description. It can be pointed out that the majority of his future courses are aimed at giving him the ability to understand and develop specific unit calculations.

The engineering decision aspects come in his ability to decide what should be studied in the first place and his ability to interpret the results of the calculations in arriving at a decision. A series of decisions results in a design, the essential function of an engineer which distinguishes him from the scientist and the technician and operator. In fact, the fundamental ideas of PACER, particularly the use of the Process Matrix and the concept of the unit calculation, could well serve as a basis conceptual framework for all of his subsequent engineering courses.

In his courses in heat transfer, fluid flow, equilibrium stage calculations, kinetics, etc. the use of the digital computer would arise naturally. Following a thorough grounding in the fundamental concepts, specific unit calculations are generally developed in detail. It would then only be required that the student be asked to write a digital computer program for the specific calculational technique just studied. This would serve as an acid test of his understanding of the material. Or, he might simply be shown the computer program and be required to use it in solving several problems. This would be done in each of his courses. Thus he would be building up a library of Equipment unit calculations, each of which he had studied in depth at the time the fundamental concepts, basic assumptions, and limitations were his primary concern. The PACER framework would serve to show where and how each course fits into the total picture and emphasize to the student the interrelationships between his courses and that what he is being taught today in fluid mechanics will be used tomorrow in design.

In the student's senior year design course he would have at his disposal a large library of unit calculations, each of which he had previously studied in depth. Emphasis could then be shifted back to consideration of the behavior of the total system. He would thus be in a position to attack a wide variety of design problems of increasing complexity. The PACER system would provide the ability to write the complex digital computer programs required for specific problems in the time required to write the Process Matrix and assemble the appropriate input data. In addition, he would be able to do a great many problems which our present seniors cannot even attempt. In this approach, the use of a digital computer would become as natural to the student as the use of his slide rule, textbooks and the library in the solution of the engineering problems he faces today and will face tomorrow.

(Authors note: Copies of the PACER program accompanied by extensive documentation for use by university staffs for educational purposes will be available in the near future. Requests should be addressed to the author at his present address.)

APPENDIX A

CH. E. 339
UNIT OPERATIONS LABORATORY

SHELL AND TUBE HEAT EXCHANGER
IDENTIFICATION NO.

A. EXPERIMENTAL CONDITIONS

1. KEROSENE	
RATE	5499690884
REYNOLDS NO.	5424397640
2. STEAM RATE	5265856000
3. WATER RATE	5466007552
PROBABLE UNCERTAINTIES	
1. KEROSENE RATE	5324056099
2. STEAM RATE	5120164178
3. WATER RATE	5315240014

B. HEAT BALANCES

	HEATER	COOLER
1. HEAT SUPPLIED BY SHELL FLUID	5562252359	-5562707174
2. HEAT ABSORBED BY KEROSENE	5560911125	5560911125
3. HEAT LOST BY RADIATION	5313457078	5225290935
4. HEAT LOST BY CONVECTION	5310412223	5217344405
5. NET HEAT LOST BY SYSTEM	5411025411	5417534137
6. SAME, AS FRACTION OF ABSORPTION	4918100816	4928786429
PROBABLE UNCERTAINTIES		
1. HEAT SUPPLIED BY SHELL FLUID	5419060794	5494464842
2. HEAT ABSORBED BY KEROSENE	5467873174	5467873172
6. FRACTIONAL HEAT LOSS	5011768355	5019285625

C. HEAT TRANSFER COEFFICIENTS

	HEATER		COOLER	
	EXPER.	PREDICTED	EXPER.	PREDICTED
1. SHELL SIDE		5422746421		5345211066
2. TUBE SIDE		5285549319		5285549319
3. FOULING		5420000000		5410000000
4. OVERALL	5291234129	5268398291	5242756633	5259197989
PROBABLE UNCERTAINTIES				
1. SHELL SIDE		5223215443		5163570229
2. TUBE SIDE		5116514913		5116514913
4. OVERALL	5210243823	5112318093	5147876196	5092899487

Note: Numbers representation
$\pm$pp nnnnnnnn = $\pm 0.\text{nnnnnnnn} \times 10^{pp-50}$
-5271200000 = -7.12
4532350000 = 0.3235×10^{-5}

THE ART OF FILTRATION
IDENTIFICATION NO. -5250000000

EXPERIMENTAL CONDITIONS

SLURRY TEMPERATURE F.	5270000000
PRESSURE DROP PSIG	5250000000

EXPERIMENTAL RESULTS

	VALUE	DEVIATION
BEST X COORDINATE	5111228350	
BEST Y COORDINATE	5330889000	5268681988
Y INTERCEPT BB	5243191675	5230745552
SLOPE KP	5323664055	5223721290
SPECIFIC CAKE RESISTANCE ALPHA	6312115680	6234477353
FILTER MEDIUM RESISTANCE RM	6172988184	6126147465
SLURRY CONCENTRATION CS	5114352000	4999056835
ALUMINUM HYDROXIDE, PERCENT	5113910505	5048642833
SILICON DIOXIDE, PERCENT	5084630350	5048639873
OPTIMUM FILTRATE VOLUME	5126604593	
OPTIMUM FRAME THICKNESS	5037837050	5013576260

Note. Numbers representation
$\pm$pp nnnnnnnn = $\pm 0.\text{nnnnnnnn} \times 10^{pp-50}$
-5271200000 = -7.12
4532350000 = 0.3235×10^{-5}

APPENDIX A

CH. E. 440
UNIT OPERATIONS LABORATORY

COOLING TOWER
IDENTIFICATION NO. 0000000002

PERIMENTAL CONDITIONS, ACTUAL

AIR	
MASS FLOW RATE GY	5417647408
TEMPERATURE F.	
INLET, DRY BULB	5278500000
INLET, WET BULB	5270500000
EXIT, DRY BULB	5271500000
EXIT, WET BULB	5270500000
HUMIDITY	
INLET	4913500000
EXIT	4915500000
WATER	
MASS FLOW RATE GX	5396623275
TEMPERATURE F.	
INLET	5271000000
EXIT	5269000000

EXPERIMENTAL AND THEORETICAL RESULTS

	THEORETICAL	ERROR	EXPERIMENTAL	ERROR
INTERFACE AREA PER UNIT VOLUME	∧		5247411960	5114190322
HEAT TRANSFER COEFFICIENT HYA	5287410747	5286395293	5322719407	5312451326
HEAT TRANSFER COEFFICIENT HY	5118436434	5118213901	5147919148	5126222804
MASS TRANSFER COEFFICIENT KYA	5335388967	5335190623	5391981407	5350409656
MASS TRANSFER COEFFICIENT KY	5174641436	5173740236	5219400465	5210616399

Note: Numbers representation
$\pm pp\ nnnnnnnn = \pm\ 0.nnnnnnnn \times 10^{pp-50}$
$-5271200000 = -7.12$
$4532350000 = 0.3235 \times 10^{-5}$

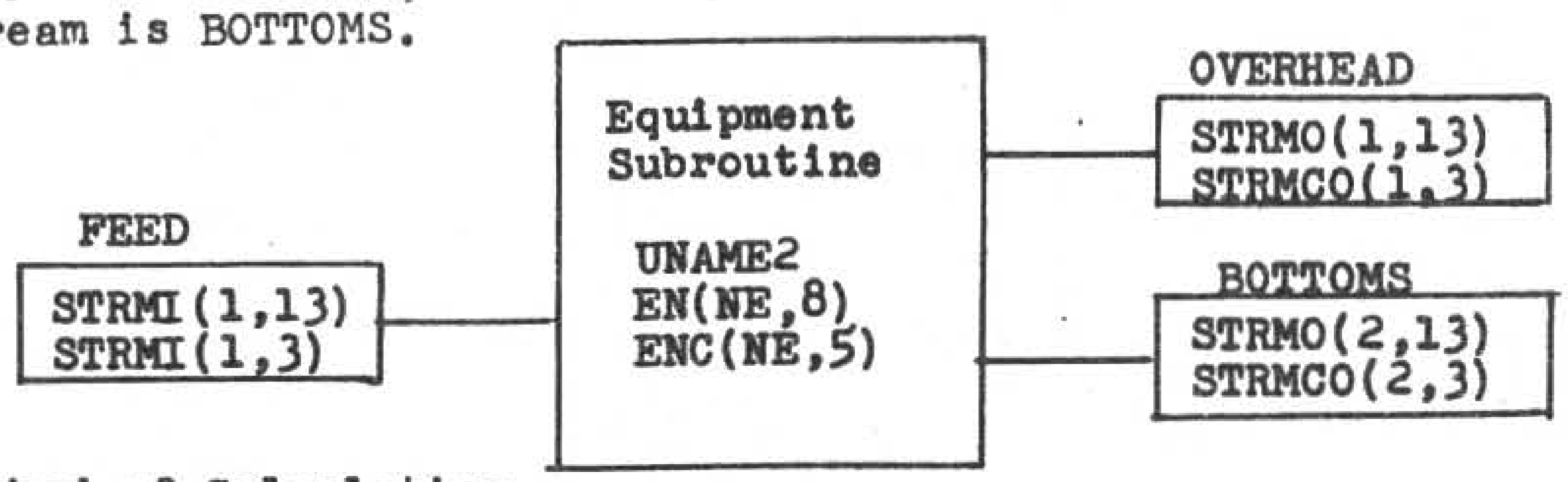
APPENDIX B. EXAMPLE SET OF EQUIPMENT SUBROUTINES AND EQUIPMENT AND STREAM MATRICES FOR USE WITH PACER

EQUIPMENT SUBROUTINE UNAME2

DISTILLATION AND REACTOR SIMULATION

This program can be used to simulate distillation towers and reactors. It is written to handle five components. One FEED stream is split into an OVERHEAD and a BOTTOMS stream.

In Process Matrix, first output stream is OVERHEADS, second output stream is BOTTOMS.



Method of Calculation

1. Calculate amounts of each component in OVERHEADS and BOTTOMS.
2. Sum component amounts to find total amount of OVERHEADS and BOTTOMS.
3. Calculate percentage compositions of OVERHEADS and BOTTOMS.
4. Calculate Fictitious Head Load.

MATRIX formats are shown on the following pages.

UNAME2 is compatible with UNAME3.

EQUIPMENT SUBROUTINE UNAME3 - COMPONENT MIXER

UNAME3 is a MIXER subroutine which adds the amounts of each component in each input stream, splits the total input amounts of each component equally between the output streams and then calculates the percentage compositions of the output streams.

STREAM PARAMETERS LIST SN and STRMI and STRMO (NS,13)

Same format required for all streams.

Matrix Row	Variable
1	Stream Number - always required
2	Stream Flag - always required
3	Total Quantity of the stream W
4	Percent of Component 1 0. - 1.0
5	Percent of Component 2 0. - 1.0
6	Percent of Component 3 0. - 1.0
7	Percent of Component 4 0. - 1.0
8	Percent of Component 5 0. - 1.0
	Σ = 1.0
9	Amount of Component 1
10	Amount of Component 2
11	Amount of Component 3
12	Amount of Component 4
13	Amount of Component 5
	Σ = W

STREAM CONTROL LIST SNC and STRMCI and STRMCO (NS,3)

Matrix Row	Variable
1	Stream Number - always required
2	Stream Flag
3	Light Key Component Number 1 or 2 or 3 or 4

Note:

Stream Type	Stream Flag
Inter-equipment	0
Feed	+1
Product	+2
Special Feed	+3

EQUIPMENT PARAMETER LIST EN(NE,8)

Matrix Row	Variable
1	Equipment Number - always required
2	Equipment Flag
3	Fictitious Head Load Q
4	Fraction of Component 1 in FEED appearing in OVERHEAD
5	Fraction of Component 2 in FEED appearing in OVERHEAD
6	Fraction of Component 3 in FEED appearing in OVERHEAD
7	Fraction of Component 4 in FEED appearing in OVERHEAD
8	Fraction of Component 5 in FEED appearing in OVERHEAD

NOTE: (1. - Fraction of Component in OVERHEAD) = Fraction of Component in BOTTOMS.

Fictitious Head Load Equation:

$$Q = A * \text{OVERHEAD} \quad B * \left(\left| F_{key} - F_{key} - 1 \right| \times 100 \right) \quad ** C$$

EQUIPMENT CONTROL LIST ENC(NE,5)

Matrix Row	Variable
1	Equipment Number - always required
2	Equipment Flag
3	1st Constant of Heat Equation (A in above equation)
4	2nd Constant of Heat Equation (B in above equation)
5	3rd Constant of Heat Equation (C in above equation)

The Use of Analog Computers in Teaching Process Control

James E. Stice and Bernet S. Swanson

Illinois Institute of Technology, Chicago

The use of indirect electronic analog computers is steadily increasing in both industry and engineering education. Such computers are used primarily for the solution of linear and non-linear ordinary and partial differential equations, and for the simulation of systems. An analog computer facility for educational purposes which is reasonably accurate and large enough to handle linear problems of moderate complexity can be obtained for $2,000. Increasingly accurate equipment with greater capacity and specialized auxiliary components requires a correspondingly larger investment.

The minimum equipment requirements include an analog computer with the necessary computing resistors and capacitors and some sort of readout device. Some of the manufacturers of small analog computers and computer components are Applied Dynamics, Inc., 2275 Platt Road, Ann Arbor, Michigan; Donner Scientific Division, Systron-Donner Corp., 888 Galindo St., Concord, California; Electronic Associates, Inc., Long Branch, N. J.; The Heath Company, Benton Harbor, Michigan; George A. Philbrick Researches, Inc., 127 Clarendon St., Boston 16, Massachusetts.

The necessary computing resistors and capacitors (if not built into the computer) may be made up by the user or they may be purchased ready to use. The resistors may be made up by attaching precision resistors having the desired resistance (one percent tolerance or better) to General Radio double plugs. Computing capacitors may be similarly made up, but it is difficult to obtain capacitors which have exactly the capacitance desired, so that trimming is almost always necessary. Further, condensors of radio quality do not have high enough leekage resistances for accurate computation purposes. Computing resistors and capacitors may be purchased from several sources, such as Donner Scientific Division of Systron-Donner Corp. and Southern Electronics Corp., 239 West Orange Grove Ave., Burbank, California. The precision capacitors are expensive.

The readout device may be an oscilloscope or some sort of recorder. A recorder is recommended for educational use, so that the student may obtain a permanent record of the solution. Recorders tested here which are quite acceptable for analog computer readout include the Brush Mark II, the Offner Type 542 and Type RP Dynographs, the Sanborn Model 1525460, the Varian G-11A, and the EAI Model 1100E Variplotter (an X-Y recorder). The July 1962 issue of Instruments and Control Systems contains a survey of 1,000 recorders.

A group of laboratory experiments follows which were developed at Illinois Tech for the process control laboratory course given in the Chemical Engineering curriculum. These experiments introduce the student to the use of the computer gradually. There are eight experiments, and a two- or three-man team of students should be able to work all of them in nine three-hour laboratory periods without too much supervision. The experiments are written for the 15-amplifier Heath Group C Computer ($945), although any computer with nine amplifiers would suffice, with the exception of the Heath Model EC-1 Computer, which can be used for only seven of the experiments.

The recorder used here was the Offner Type 542 Dynograph ($1,145). This is a two-channel, galvanometer-type recorder which is also used for a great variety of other applications around the department. If money is tight, excellent results should be obrained with a single-pen, potentiometer-type recorder such as the Varian G-11A recorder (base price with Type B-1 Input Chassis is $540). The range of applications of such recorders is not as broad as that of the galvanometer-type recorders, since the potentiometric recorders cannot be used for signals having frequencies above one cycle per second. However, the equations can be time-scaled so that good results can be obtained with these recorders.

The computing capacitor requirements for these experiments are: one 0.01 ufd, one 0.1 ufd, and four 1.0 ufd capacitors. Resistors required are: two 0.1 megohm, one 0.2 megohm, one 0.4 megohm, four 0.5 megohm, eight 1.0 megohm, two 2.0 megohm, one 5.0 megohm, and one 10.0 megohm resistors (one percent tolerance or better).

In addition to the equipment already specified, the second part of Experiment Two requires a diode function generator (DFG) for the generation of the valve characteristic curve. There are a number of diode function generators on the market, but by far the least expensive is the Model ES-600 (kit) manufactured by the Heath Company ($72.95). This DFG provides only ten straight line segments, and it is understandably less accurate than a DFG costing $450, but it is adequate for instructional purposes and the price is attractive.

These experiments have worked out very well; the students have learned how to solve relatively simple process control problems on the computer. There have, in addition, been some bonus results which were not anticipated when the program was begun. Students tend to get rather disconcertingly enthusiastic about the computer after they begin to understand how to operate it, and they leave the laboratory only after repeated threats of bodily harm. They concoct their own problems and return to the laboratory on their own time. Further, they experience a renewed interest and a more mature understanding for differential equations, and the electrical engineering department reports that our students are badgering their staff to give them more electronics. This awakening of intellectual curiosity in nearly all the students who have worked with the computer has been a delightful, if somewhat wearing, experience for the staff members who teach this laboratory.

Building an analog computer laboratory poses the problem of how many students can be handled at one time. A team should consist of no more than three students, and two is better. A class of twelve students may thus imply four to six computers, and this becomes an expensive operation. The Donner Model 3500 and Model 3400 computers have removable problem boards, as do many of the larger computers (Applied Dynamics, Electronic Associates, Berkeley Division of Beckman Instruments). Also, Prof. James O. Osburn of the Chemical Engineering Department, State University of Iowa, Iowa City, has devised a plugboard for use with the Heath Group C computer. This plugboard has connections to four of the computer amplifiers, three of the initial condition power supplies, and a 100-volt and a -100-volt supply. Each plugboard also contains four integral coefficient potentiometers. A plugboard costs about $15.50 to make. A team of students can patch up a problem on their own plugboard and when wiring is completed the board is attached to the computer and the solution can be run off in a short time. In this way one computer can serve a class of perhaps ten to twelve students. Osburn's plugboard is described in the Journal of Chemical Education, 38, 492 (1961), and further details may be obtained direct from Dr. Osburn.

The development of these experiments and the manual which accompanied them was supported by a National Science Foundation grant.

EXPERIMENT ONE

INTRODUCTION TO THE HEATH GROUP C ELECTRONIC ANALOG COMPUTER

This experiment is intended to familiarize the student with the basic techniques of analog computation on the Heath Group C electronic analog computer. After the various mathematical operations which the computer can perform have been studied, they can be utilized to solve a classical rpoblem in physics, such as the body falling freely in a vacuum from a position of rest.

EXPERIMENT TWO

FUNCTION GENERATION

PART I: Use of the Computer to Generate Functions

An analog computer can be used to generate a variety of functions for use as problem inputs.

PART II: Use of the Diode Function Generator

EXPERIMENT THREE

COMPUTER SOLUTION OF LINEAR SECOND-ORDER DIFFERENTIAL EQUATIONS

Ordinary linear differential equations occur commonly in science and engineering. The examples used here will be limited to differential equations with constant coefficients, so that function multipliers will not be required. A classical problem in mechanics is the mass-spring-damper system, in which a mass is supported by a spring and a dashpot.

EXPERIMENT FOUR

FREQUENCY RESPONSE DIAGRAM FOR A MECHANICAL SYSTEM

In the previous experiment, the analogy between the mass-spring-damper system and the time-scaled R L C circuit was developed, and the transient response of these systems was studied for the case with no forcing function applied. In this experiment, the mechanical system will be forced to oscillate by impressing a sinusoidal forcing function on the mass. The system will be subjected to forcing functions of different frequencies, and a frequency response diagram will be constructed for the system.

EXPERIMENT FIVE

COMPUTER SIMULATION OF SYSTEM COMPONENTS

One of the most important applications of analog computers is the simulation of physical systems. The computer is programmed to solve the differential equation or set of equations which represent the system. When this has been done, it turns out that certain portions of the computer circuit represent identifiable parts of the physical system under study, so that it becomes natural to think of these circuit components as though they were the corresponding components of the physical system. This will be illustrated in the several parts of this experiment.

PART I: Single-Tank Liquid Flow Process

This is illustrated by a liquid flow process in which liquid flows into a tank at a rate of $F_o(t)$ cubic feet per minute, and flows out at a rate of $F_1(t)$ cubic feet per minute. The capacity of the tank is C_1 cubic feet of liquid per foot of depth, which is numerically equal to the cross-sectional area of the tank in square feet. To keep the problem simple, assume that the cross-sectional area of the tank is uniform from top to bottom, as would be the case with a vertical cylinder or a rectangular tank. The head of liquid in the tank is h_1 feet. The

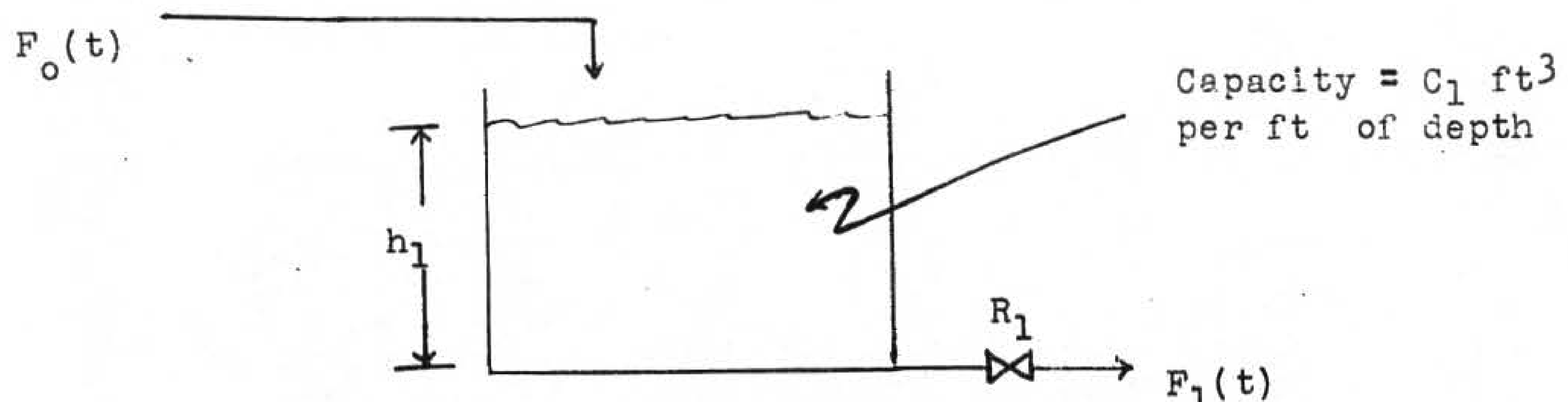


Figure 5-1: Single-Tank Liquid Flow Process

liquid flowing out of the tank suffers head losses due to contraction and expansion, and friction in the piping and fittings. All of these factors are lumped into one equivalent resistance term which is designated R_1 (foot) (minutes) per cubic foot of flowing liquid. This equivalent resistance is equal to the slope of the head versus flow curve <u>in the region of interest</u>. This curve is not normally linear, but it may be <u>approximately linear in the</u> region of interest. Again in the interests of keeping the problem simple, the curve which relates head to flow F_1 will be assumed to be a straight line.

PART II: A Second Single-Tank Liquid Process

A second tank will be simulated, the new tank being similar to the first. The process time constant will be different, since the second tank will have a larger cross-sectional area (capacity) and a somewhat different equivalent resistance. As before, capacity and resistance will be assumed constant. The flow into Tank 2 will be $F_1(t)$ cubic feet per minute, and the flow out will be $F_2(t)$ cubic feet per minute.

PART III: Two-Tank Liquid Flow Process

Tank 1 is the tank of Part I and Tank 2 is the tank of Part II. It is desired to know what the transient response of this system will be if a step input is applied suddenly increasing flow $F_o(t)$. When the transient response curve has been obtained, it will be used to calculate the time constants of the process.

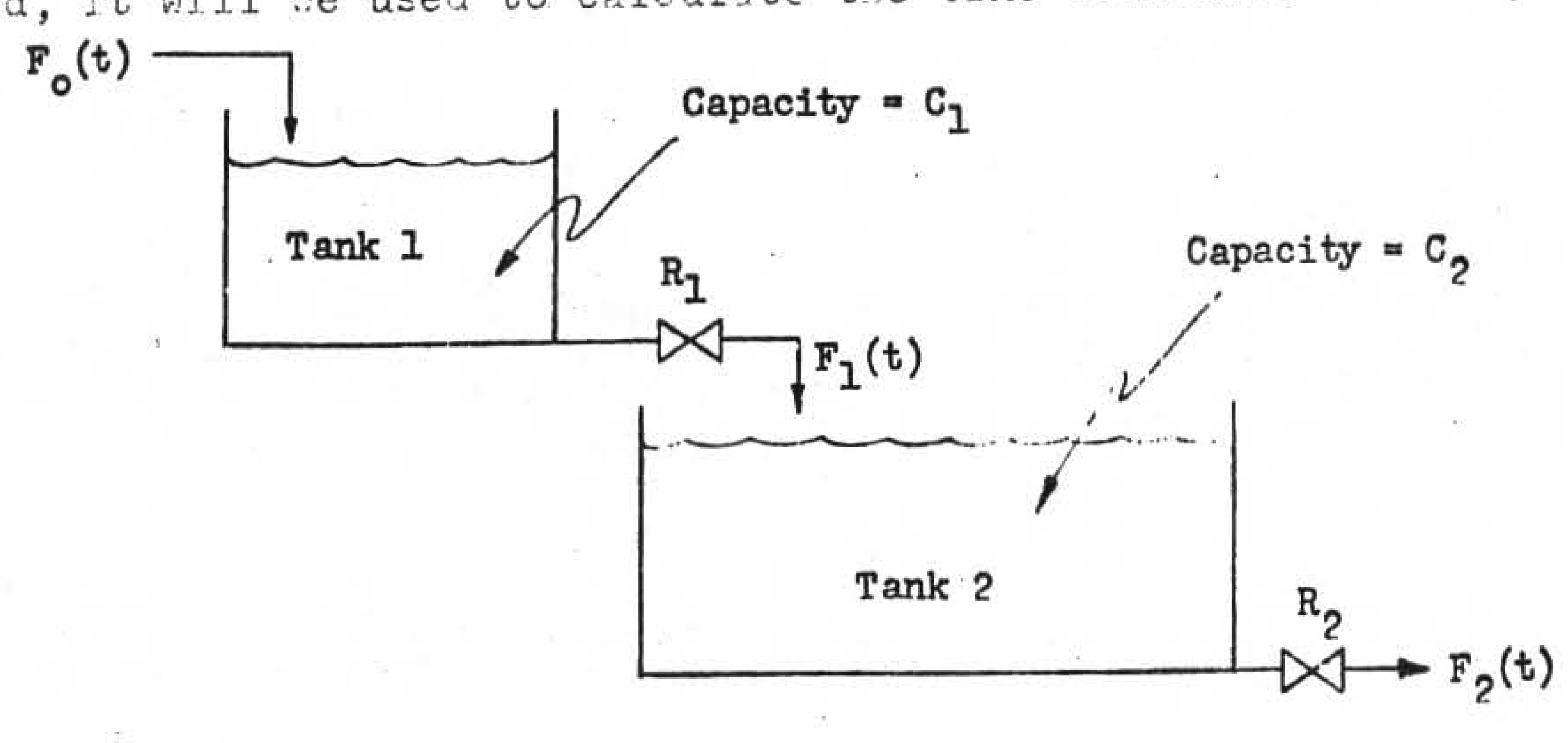


Figure 5-3: Two-Tank Liquid Flow Process

EXPERIMENT SIX

OPEN-LOOP RESPONSE OF PROCESS

The two-tank process of Experiment Five is to be instrumented to maintain a flow rate of twenty gallons per minute for flow $F_2(t)$. The control valve and the flow sensing device will be simulated in this experiment and the open-loop response of the system will be studied. In Experiment Seven the pneumatic controller will be studied. The controller will be added to the rest of the equipment in Experiment Eight, and the closed-loop behavior of the entire system will be observed for various controller settings.

Flow $F_2(t)$ out of Tank 2 will be maintained constant by controlling flow $F_o(t)$ into Tank 1. Flow $F_L(t)$ is an intermittent stream which also flows into Tank 1 at arbitrary intervals and for varying lengths of time. To simplify the analysis, it will be assumed that the nature of the process is such that Tank 1 will never run dry or overflow.

EXPERIMENT SEVEN

COMPUTER SIMULATION OF A PNEUMATIC CONTROLLER

In order to control the system in the previous experiment, some sort of controller is necessary. This controller receives the air pressure signal from the flow sensing device, subtracts this signal from the set point signal to produce an error signal, and acts upon the error signal to reposition the control valve.

Pneumatic controllers can be obtained with up to three modes of control, these being proportional, derivative (also called rate or preact), and integral (also called reset rate) modes. The particular process under study can be controlled very nicely by a controller having proportional and integral action, and this is the type of controller which will be simulated.

EXPERIMENT EIGHT

BEHAVIOR OF THE CLOSED-LOOP PROCESS

The complete process of Experiment 6 will be simulated and the effect of various controller settings will be observed. The pertinent facts about the process are summarized below.

(1) It is desired to maintain a constant flow of twenty gallons of liquid per minute out of the bottom of the second of two non-interacting, series-connected tanks. This flow is designated $F_2(t)$. To accomplish this a pneumatic control valve regulates the flow of liquid into the top of the first tank, the regulated stream being $F_o(t)$.

(2) In addition to $F_o(t)$ there is an intermittent stream, $F_L(t)$, which also flows into the top of the first tank. This flow is unregulated, and occurs in varying amount and on no regular schedule. The amount of this flow is small compared to $F_o(t)$.

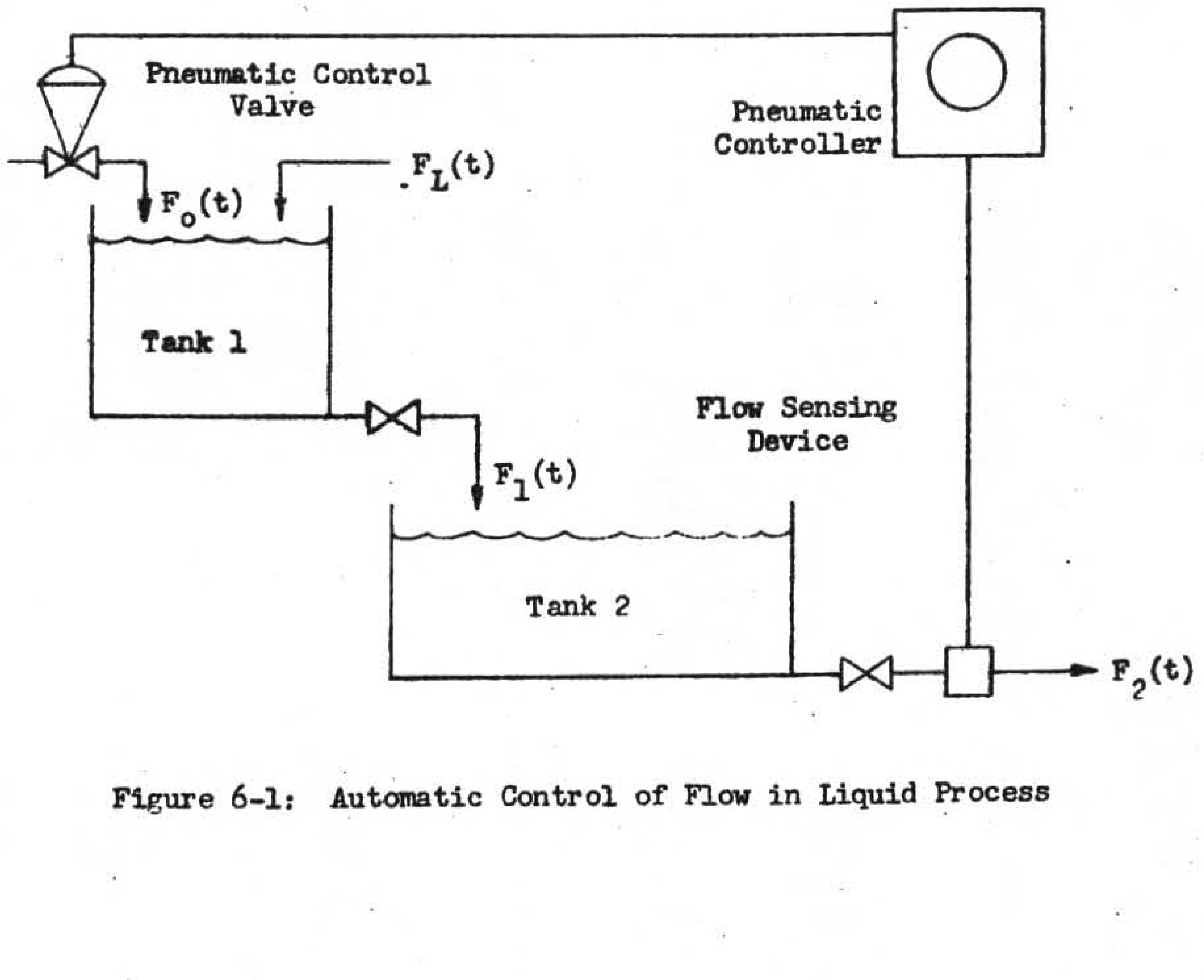


Figure 6-1: Automatic Control of Flow in Liquid Process