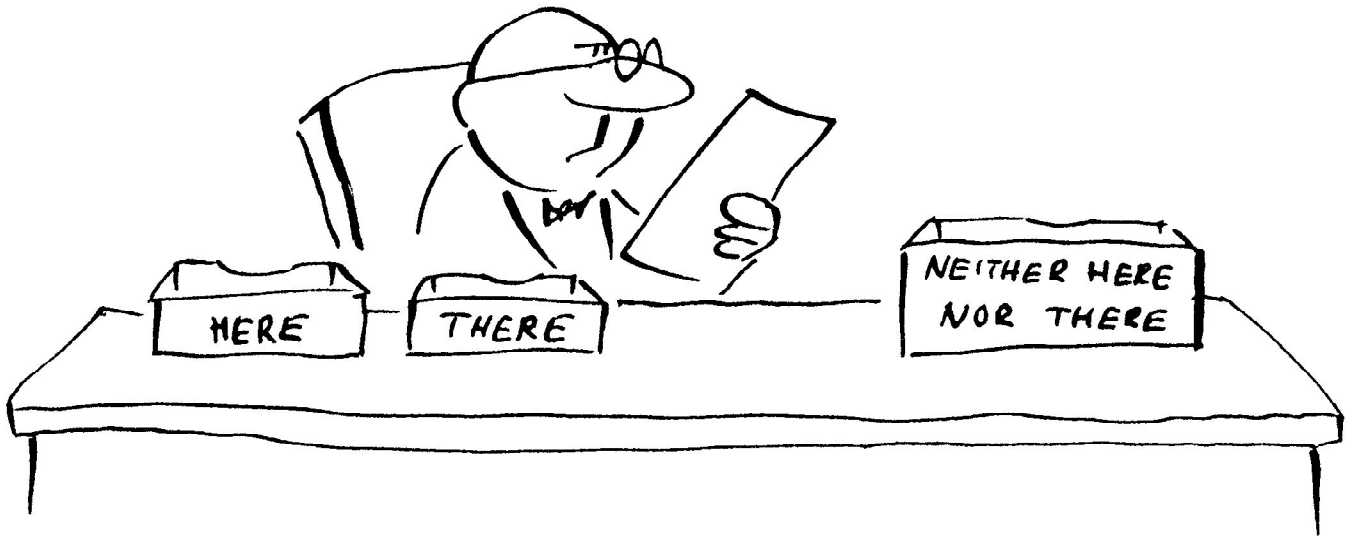
CACHE NEWS

NEWS ABOUT COMPUTERS IN CHEMICAL ENGINEERING EDUCATION.

No. 25

Fall 1987


SURE I STORED IT IN MEMORY. WHAT WAS IT?


HERE
THERE
NEITHER HERE NOR THERE

WHAT IS CACHE?

CACHE is a not-for-profit organization whose purpose is to promote cooperation among universities, industry and government in the development and distribution of computer-related and/or technology-based educational aids for the chemical engineering profession.

CREATION OF THE CACHE CORPORATION

During the 1960s the rapid growth of computer technology challenged educators to develop new methods of meshing the computer with the teaching of chemical engineering. In spite of many significant contributions to program development, the transferability of computer codes, even those written in FORTRAN, was minimal. Because of the disorganized state of university-developed codes for chemical engineering, fourteen chemical engineering educators met in 1969 to form the CACHE (Computer Aids for Chemical Engineering) Committee. The CACHE Committee was initially sponsored by the Commission on Education of the National Academy of Engineering and funded by the National Science Foundation. In 1975, after several successful projects had been completed, CACHE was incorporated as a not-for-profit corporation in Massachusetts to serve as the administrative umbrella for the consortium activities.

CACHE ACTIVITIES

All CACHE activities are staffed by volunteers including both educators and industrial members and coordinated by the Board of Trustees through various Task Forces. CACHE actively solicits the participation of interested individuals in the work of its ongoing projects. Information on CACHE activities is regularly disseminated through **CACHE NEWS,** published twice yearly. Individual inquiries should be addressed to:

CACHE Corporation
P.O. Box 7939
Austin, Texas 78713-7939
(512) 471-4933

CACHE TRUSTEES (and their BITNET ID)

President: Gintaras V. Reklaitis, Purdue University, GVR@PURCHE

Vice President: H. Scott Fogler, University of Michigan

Secretary: John C. Hale, DuPont

Executive Officer: David M. Himmelblau, University of Texas at Austin

Academic Trustees: Yaman Arkun, Georgia Institute of Technology
L. T. Biegler, Carnegie-Mellon University
Brice Carnahan, University of Michigan
Michael B. Cutlip, University of Connecticut, MCUTLIP@UCONNUM
Michael F. Doherty, University of Massachusetts
Thomas F. Edgar, University of Texas at Austin
Bruce A. Finlayson, University of Washington
Ignacio Grossmann, Carnegie-Mellon University
Richard S. H. Mah, Northwestern University, RMAIL001@NU
Duncan A. Mellichamp, University of California, Santa Barbara
Manfred Morari, Califonia Institute of Technology
Peter R. Rony, Virginia Polytechnic Institute and State University, RONY@VTVM1
Stanley I. Sandler, University of Delaware
J. D. Seader, University of Utah, SEADER@CCA.UTAH.EDU
Warren D. Seider, University of Pennsylvania
John H. Seinfeld, California Institute of Technology
George Stephanopoulos, Massachusetts Institute of Technology

Industrial Trustees: J. J. Haydel, Shell
Norman E. Rawson, IBM, RAWSON@MILVM1*
Edward D. Rosen, Monsanto Company
Jeffrey J. Siirola, Eastman Kodak
H. Dennis Spriggs, Linnhoff March
Joseph D. Wright, Xerox, WRIGHT@MCMASTER or WRIGHT.XRCC-NS@XEROX.COM

*Gateways must be opened at both ends of the BITNET link.

Advisory Committee: Lawrence B. Evans, Massachusetts Institute of Technology
James R. Fair, University of Texas at Austin
Thomas H. Laffrerre, Monsanto Company
W. Robert Marshall, Jr., University of Wisconsin
John J. McKetta, University of Texas at Austin

CACHE NEWS

The **CACHE News** is published twice a year and reports news of CACHE activities and other noteworthy developments of interest to chemical engineering educators. Persons who wish to be placed on the mailing list should notify CACHE at the aforementioned address. Contributions from CACHE representatives are welcome. This issue was edited by D. M. Himmelblau with contributions from a number of CACHE members and representatives.

CACHE TASK FORCES AND COMMITTEES

STANDING COMMITTEES

Publications

Prof. Brice Carnahan
Dept. of Chemical Engineering
University of Michigan
Ann Arbor, Michigan 48109
(313) 764-3366

Newsletter

Prof. David M. Himmelblau
Dept. of Chemical Engineering
University of Texas
Austin, Texas 78712
(512) 471-7445

Development

Dr. H. Dennis Spriggs
Linnhoff March
P.O. Box 2306
Leesburg, Virginia 22075-7617
(703) 777-1118

TASK FORCES

Conferences

Prof. Richard S. H. Mah
Dept. of Chemical Engineering
Northwestern University
Evanston, Illinois 60201
(312) 491-5357

Large-scale Systems

Dr. Jeffrey Siirola
Eastman Kodak Co.
P.O. Box 1972
Kingsport, Tennessee 37662
(615) 229-3069

Case Studies

Prof. Manfred Morari
Dept. of Chemical Engineering
206-41
California Institute of Technology
Pasadena, California 91125
(818) 356-4186

MicroCache

Prof. Brice Carnahan
(see above)

Curriculum

Prof. Warren Seider
Dept. of Chemical Engineering
University of Pennsylvania
220 S. 33rd Street/D3
Philadelphia, Pennsylvania 19104
(215) 898-7953

Electronic Mail

Prof. Peter Rony
Dept. of Chemical Engineering
Virginia Polytechnic Institute &
State University
Blacksburg, Virginia 24061
(703) 961-7658

AD HOC COMMITTEES

Laboratory Applications of Microcomputers

Prof. Duncan Mellichamp
Dept. of Chemical Engineering
University of California
Santa Barbara, California 93106
(805) 961-2821

Expert Systems

Prof. George Stephanopoulos
Dept. of Chemical Engineering
Massachusetts Institute of Technology
66-562
Cambridge, Massachusetts 02139
(617) 253-3904

PID Control Simulation Using LabVIEW

by Glenn D. Munkvold,
Gerald G. McGlamery &
Kevin J. McLaughlin

The University of Texas at Austin

The major difficulty in constructing an interface between a microcomputer and laboratory instrumentation for the purposes of data acquisition and process control lies in the development of device drivers for the microcomputer interface port. Typically a researcher must program these drivers or pay a programmer for this service. In addition, this task is further complicated by the need to write these device drivers in a language which is fast, and therefore compiled or assembled, and which allows low-level access to the microprocessor and its memory. Well-known languages such as FORTRAN and BASIC must therefore be discarded in favor of more esoteric languages such as C.

In response to these problems National Instruments has developed a graphically-oriented programming system for data acquisition and process control called LabVIEW (Laboratory Virtual Instrument Engineering Workbench). LabVIEW was designed for the Macintosh to provide a software environment for creating device drivers of almost any configuration. The Macintosh user interface that uses pulldown menus and multiple windows, allows the development of a virtual instrument control panel (the PANEL view, Figure 1) and a block diagram representation of the device driver program (the DIAGRAM view, Figure 2). LabVIEW allows direct interfacing to instruments which use the RS-232 protocol and indirect interfacing to GPIB (IEEE-488) instruments and IBM PC-AT I/O expansion boards through National Instruments' GPIB-MAC and MacBus interface hardware.

In addition to data acquisition and process control, LabVIEW is also well-suited to producing simulated virtual instruments. The input and output controls of the LabVIEW PANEL allow an unfamiliar user to intuitively operate the simulation without understanding its structure. However, should the user wish to investigate the structure of the simulation, the LabVIEW DIAGRAM provides an intuitive representation of this structure.

LabVIEW Description

General descriptions of LabVIEW have been provided elsewhere (Wolfe, 1986; Vose and Williams, 1986); therefore, only a few basic concepts will be discussed.

LabVIEW can be considered a complete programming language. It provides the three basic language constructs necessary to develop computer algorithms: case structure, iterative loop structure, while loop structure. These program structures are represented graphically in LabVIEW. LabVIEW also provides one other important structure, a sequence structure which forces certain parts of a LabVIEW program to be executed before other parts. Because LabVIEW is inherently parallel in execution, a distinct advantage in many applications, the sequence structure can be very important for such tasks as instrument initialization and data analysis.

Several useful programming concepts have also been implemented in LabVIEW. Both loop structures contain shift registers which recirculate variable values from the end of one loop execution to the beginning of the next. These variables can be either scalars or arrays. LabVIEW also allows dynamic memory allocation. Only the dimensionality of an array must be declared; LabVIEW will determine the actual array size.

The final and perhaps most important concept in LabVIEW is the virtual subinstrument. Each instrument that a user constructs may be saved as a subinstrument and given an icon. This subinstrument can then be used in the construction of other instruments. Each of the various numeric and string operations provided in LabVIEW are actually subinstruments. This concept allows the user to program in a structured manner.

Objectives

The objectives of this project were twofold. The first objective was to examine the utility of LabVIEW for developing interactive simulations. The second objective was the development of a useful tool for undergraduate instruction in process control. In designing this

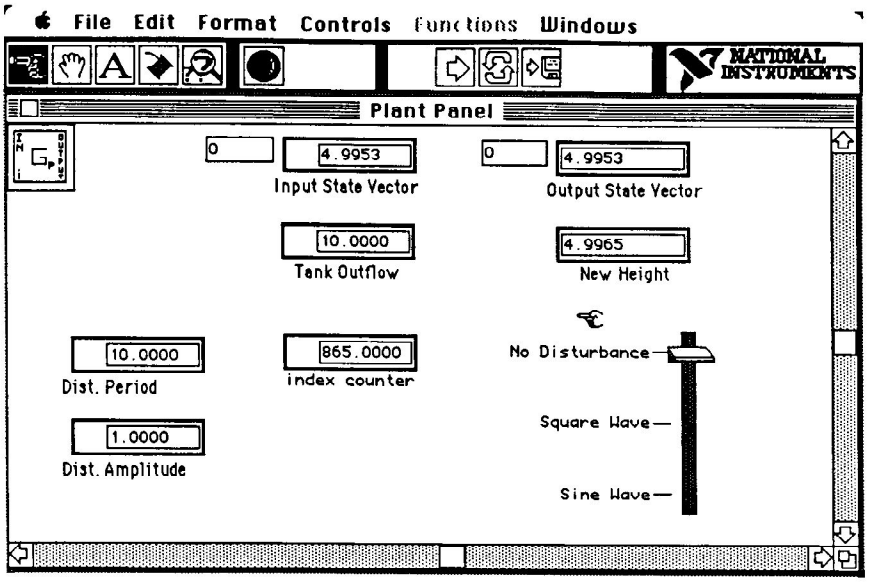


Fig. 1: Typical LabVIEW panel.

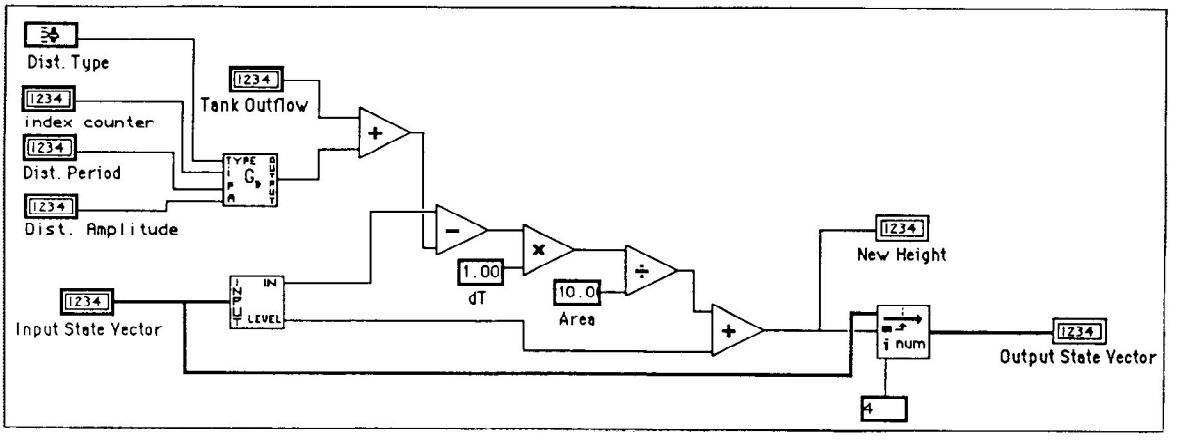


Fig. 2: Typical LabVIEW diagram.

tool, the following criteria were established:

1. The simulation must be interactive. It should provide the student with instant feedback on the effect of changes to the simulator controls.
2. The simulation must represent a true digital controller.
3. The simulation should provide a useful laboratory experience to the student. A multitude of variables should prevent the simulation from becoming too predictable.
4. The simulation program should be constructed so that advanced students can program their own control algorithms.
5. Advanced students should acquire the skills to program device drivers for real applications.

Simulator Description

A simple liquid level control simulator was written with LabVIEW (version 1.02). The level in a single tank (with a nominal outlet flow rate) was maintained by manipulating the inlet flow rate. The tank and measurement elements are modeled using first-order transfer functions with no time delay. The final control element (a valve on the inlet to the tank) is linear. Disturbances in load (outlet flow rate) can be sine wave or square wave with varying amplitude and frequency. A PID controller was implemented using a velocity algorithm with rectangular integration and backwards difference derivative. Feedback is through a state vector circulated back in a while structure using shift registers. The shift register is the only construction in LabVIEW that allows the results from calculations in a loop to be returned and run through the loop again.

LabVIEW inherently satisfies the first criterion above. In order to satisfy the second, a zero-order hold was simulated to allow students to study the relationship between sample time and controller performance. Measurement properties (gain and time constant) as well as disturbance characteristics (amplitude and frequency) can be manipulated to create a large number of different exercises for class use. Different plans can be created and implemented easily. Furthermore, the structured nature of LabVIEW allows a user to easily include a different controller.

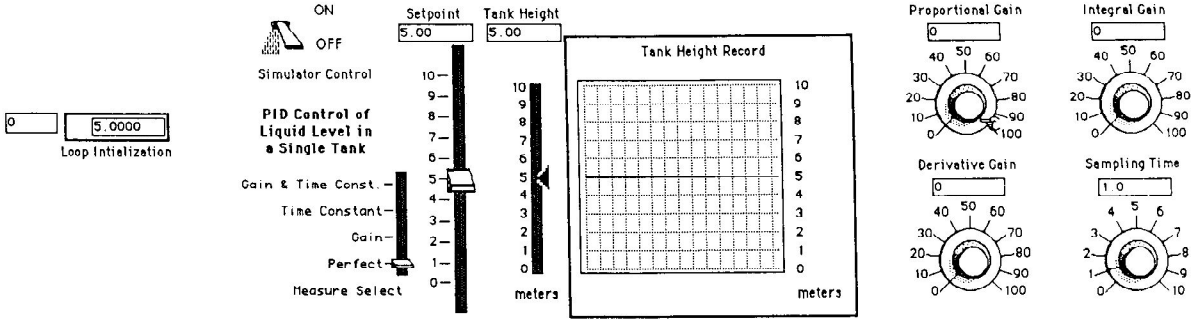


Fig. 3: Simulator panel.

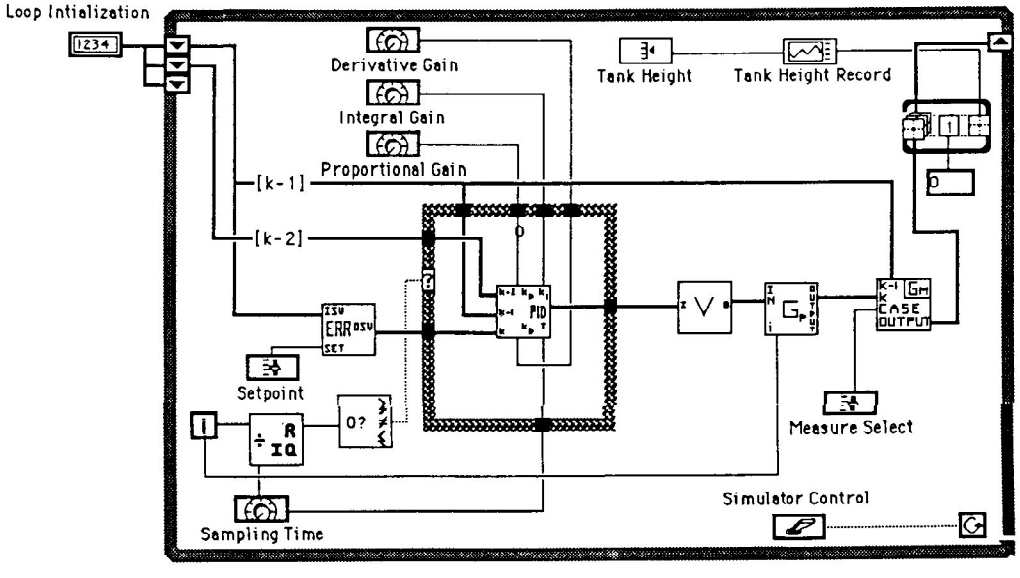


Fig. 4: Simulator diagram.

Figure 3 shows the simulator front panel. Many simulation options are controlled from this panel. The state vector is initialized at far left in the Loop Initialization control. Unfortunately, as a default LabVIEW requires all values in a vector to have the same value. Therefore the state vector must be manually initialized before running. The Simulator Control toggles a logical variable that controls execution of the while loop in which the entire simulation runs. Measure Select determines how the measurement model performs, Perfect gives perfect measurement, Gain applies a gain to the measurement, Time Constant uses unity gain but gives a time constant (and hence first-order behavior) to the measurement, and Gain & Time Const. applies both cases above to the measurement. The user specifies the set point on the Set Point control; tank height, in meters, is reflected in both the Tank Height indicator and the Tank Height Record (which operates like a strip chart). Controller parameters and sampling time are manipulated with the dials at the right.

All controls are manipulated with mouse movements after the operation icon (a left pointing index finger) has been selected from the palette in the upper left portion of the screen (Figure 1).

Figure 4 shows the diagram associated with the simulator panel. The diagram is where a program is written; the panel is primarily an input/output device. When controls are created on the front panel, small icons representing those controls appear on the diagram. Other icons represent available operations and functions within LabVIEW (such as math, string and array operations, special constants, user-defined constants, timers, comparative operations and input/output structures), as well as user-constructed subinstruments. LabVIEW also provides the user with the ability to use standard computer code (in C) as a subinstrument.

All of the elements represented as icons have associated "connector panes" that determine how information is to be passed to the element. All data are typed, and when elements are wired together (using the wire spool icon in the palette) the types are checked to ensure compatibility. Incompatible types are flagged as "bad wires" and displayed as dashed lines. Other line types indicate the type of information being passed, e.g., logical, graphical, scalar, vector, etc. The icons labeled ERR, PID, V, G_2 and G_M are user-defined subinstruments for the error junction, the PID controller, the final control element (a valve), the plant, and the measurement device, respectively. These subinstruments are constructed from other subinstruments; there are twelve subinstruments in the simulator (Figures 1 and 2 are from the plant subinstrument).

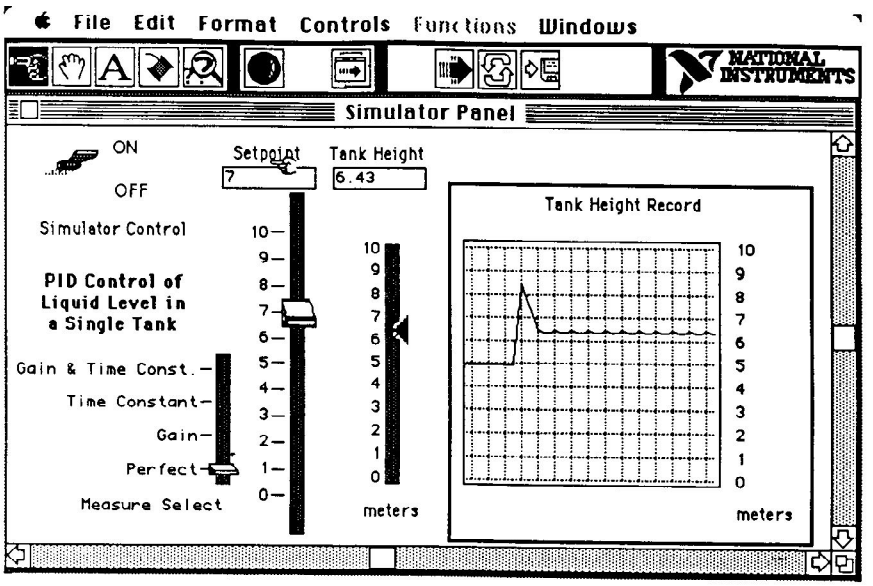


Fig. 5: Proportional control; gain = 35, sample time = 1 second.

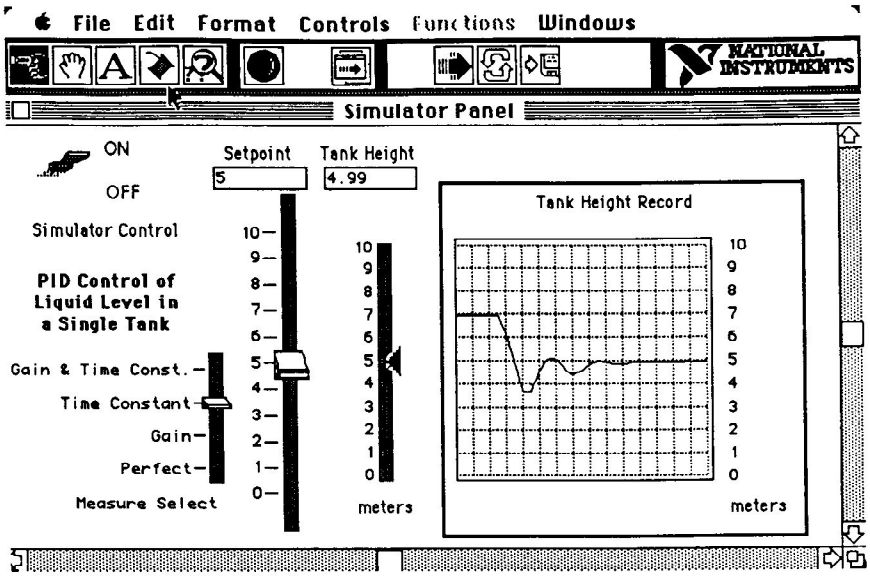


Fig. 6: Measurement dynamics.

The arrows on the edge of the while loop are the shift registers (used here to feed back the state vector); the box with a wide border inside the while loop is a case structure. By default, the case structures in LabVIEW accept any number of cases. The case structure around the PID controller, however, has been converted to accept the logical output from the "0?" comparator.

The zero-order hold is implemented using the while loop counter (the lower case "i" in a box). The counter value is divided by the sampling time and the remainder compared to zero. If it is not equal to zero, the comparator output is true and Case 1 (no control) is selected. Otherwise, comparator output is false and the controller case is selected.

Simulator Operation

The simulator is operated through the use of the front panel, where parameters can be manipulated while viewing the process output in real-time, in this case a strip chart of tank level vs. time. Real-time in the LabVIEW environment is actually based on simulation time steps, the screen being updated each time calculations are made throughout the simulator. The simulation was run on a Macintosh Plus computer with hard disk. (LabVIEW is not copy-protected.) Time to start up the simulator was significant, approximately three minutes, but execution was not excessive.

The plant is designed to control the liquid level in a tank by manipulating input liquid flow rate in response to a level set point. Average output flow rate is set by the user. Simulations can be run to show both servo (set point) and load (disturbance) responses within the system. Two types of disturbances have been built into the system, each with variable amplitude and period. The disturbances are additions to the output flow rate (simulating variable demand) in the form of either a sine wave or square wave. These disturbances are set behind the front panel so that they can be hidden from a student in the case of classroom operation. The user can tune the system for optimal response to these disturbances by adjustment of the PID controller parameters on the front panel.

Figures 5 and 6 summarize results for a series of simulations with LabVIEW. The single tank simulator was initialized at a steady height of five meters. Tank area was ten square meters; the nominal demand on the tank was 10 cubic meters per second (one iteration of the outermost loop was assigned to be one second). Tank height is controlled with a valve on the inlet to the tank. Figure 5 shows a step change from five to seven meters using proportional control. The gain was 35 and the sample time was one second (note offset is present). Figure 6 shows the effect of measurement dynamics during a step change (no disturbance, PI control, gain = 20, reset rate = 5, sample time = 1 second).

Conclusions

As a teaching tool, LabVIEW shows excellent potential. The simulator interface is easy to master, and little if any knowledge of the underlying programming is needed to run the simulator. As stated earlier, programming is relatively intuitive, making extensions of a given problem easy to implement either by students or instructors. In addition a large number of process and control parameters can be studied. Plant variables (in this case tank diameter and outlet flowrate) can be easily changed. Process control variables such as proportional gain, integral time, derivative time and sample time can be changed in the LabVIEW real-time environment. Process disturbances and measurement errors (incorrect gain and time constant) are also readily accessible to the user.

LabVIEW is a versatile, easy-to-learn piece of software with a wide range of uses in chemical engineering and chemical engineering education. With its intuitive, icon-oriented style of programming, LabVIEW represents a good implementation of the Macintosh user interface. While LabVIEW was used as a simulator in this study, its best utility is probably as a controller-monitor for GPIB instruments using the available hardware from National Instruments. Icon-oriented programming combined with special functions for input/output manipulation makes creation of device drivers relatively easy.

Many drivers already exist and are available from National Instruments. However, caution should be exercised in application to process systems with fast dynamics. When the dynamics of a system are faster than the Macintosh's ability to run through the necessary calculations, system performance will degrade.

The learning curve of LabVIEW is quite sharp. The programming style is, as stated earlier, intuitive, making it simple to browse through available functions and experiment with their use. A novice user can produce working simulations in a short time with the aid of the on-line help system and LabVIEW documentation. The documentation for LabVIEW is complete although somewhat cumbersome. The answers to most questions are generally available, although they can be difficult to find within the documentation.

References

1 Wolfe, R., ***Electronic Design,*** 34, April 17, 1986.

2 Vose, G. M., and G. Williams, ***BYTE,*** 11 (9), 1986.

COMPUTING IN CHEMICAL ENGINEERING EDUCATION

by: Thomas F. Edgar
The University of Texas at Austin

The significant changes in computer technology during the past fifteen years are having a major impact on how chemical engineering undergraduates are trained. In response to accreditation guidelines, chemical engineering curriculum should now include use of computation and solution of open-ended problems in all courses, not just in the traditional process design and control courses. This paper discusses recent trends in formulation and implementation of degree requirements which impact computing; it also covers several activities in new software development for use by undergraduates in chemical engineering.

INTRODUCTION

The most rapidly changing technology in the world during the past decade has been computer technology. The personal computer has now replaced hand calculators as the standard engineering tool. The capabilities of the modern personal computer have also superceded those of a typical 1970 mainframe costing 50 times more and occupying a large room.

Currently available computers are indeed multifunctional; modular hardware and software upgrades can convert a modest PC into a system with full graphics capability, which can communicate with other computers in a network, able to simulate process flowsheets, carry out data acquisition and even perform process control. In the late 1970s there were very well-defined equipment classes for each function: you needed a special terminal for interactive graphics as well as for preparing charts and figures, a mainframe for computer simulation and a special minicomputer for data acquisition and control. Time and technological improvements have blurred these distinctions; the modern engineering workstation has evolved.

The availability of inexpensive, ubiquitous personal microcomputers and supermicrocomputers is having a profound influence on the way the professional chemical engineer performs his or her job. Similarly, universities are attempting to provide an up-to-date computing environment for the training of chemical engineers, although clearly there is a tremendous need to develop new software which is oriented towards the education of the chemical engineering undergraduate.

In this paper educational trends such as accreditation guidelines which are pertinent to computing technology are identified. One fact is clear: the whole field of chemical engineering is undergoing significant change and computing is a pervasive part of the picture. Recent developments in such areas as computer graphics, communication, computerized data acquisition, numerical analysis, simulation, optimization, artificial intelligence and databases impact upon the teaching process. Given the large amount of activity in computing and the limited time and resources, no department by itself can address the computing problem. Cooperative efforts such as those fostered by CACHE will be required to modernize the curriculum and pedagogy of most departments.

EDUCATIONAL TRENDS

AIChE is the organization that accredits various departments of chemical engineering in the United States through its Education and Accreditation Committee. Over the past five years, AIChE has taken a progressively stronger stance on the expected competence in computing, requiring instruction in process control and strongly encouraging that advanced interactive computation systems be made accessible to and employed by engineering students. Formal course instruction in computational skills is desirable, either within or without the department, and integration of machine computation into every course possible should be undertaken.

A related issue in accreditation is the teaching of engineering design, which is supposed to comprise one-eighth of the total curriculum. Design involves the integration of engineering facts and principles in the solution

of comprehensive engineering problems or student projects. Such problems should be sufficiently open-ended, with multiple paths to obtain a number of feasible answers, and should pay attention to process economics. While a "capstone" senior design course must certainly be taught, additional design credits can be earned in courses on fluid mechanics, heat and mass transfer, reactor design and process control. Denn[1] has argued that design is not a subject which demands different treatment from engineering science; rather there should be a design component in every course. However finding a way to incorporate meaningful open-ended problem solving into every course is a more difficult goal. Computing technology offers new opportunities for open-ended problem solving through integrated lessons that enable students to explore design implications of engineering science.

Computing also offers a way to cover more material in the highly-constrained four-year B.S. chemical engineering program that must be taught in the U.S. In order to avoid cutting back on content, computing can be used to provide enrichment and enhanced capabilities for the student. For example, manual construction of engineering graphs is tedious, time-consuming and has limited educational value; therefore, preparation of such graphs on the computer is very attractive and efficient for teaching and reinforcing the concepts. With tested software available in a convenient form (diskettes), students spend less time writing their own programs and are able to concentrate on the proper selection of a computer program and its application. While the computer carries out such tasks as repetitive calculations and plotting of results, the student learns how to use the computer as a tool, integrating information and optimizing decisions. The microcomputer therefore allows the student to focus on the interplay of important equations and the effect of changes in the problem of specifications.

A study on changes in the curriculum of chemical engineering was recently completed by the Septenary Committee at the University of Texas. This committee, composed of industrial leaders from a wide variety of companies that employ chemical engineers, made extensive recommendations on how the teaching of chemical engineering should be changed so that the B.S. chemical engineer is better prepared for the competitive environment in today's workplace.[2]

Those recommendations pertinent to computing are paraphrased below:

1. Although students should develop reasonable proficiency in programming, the main thrust should be the use of standard software including the merging of various programs to accomplish a given task. Major emphasis should be on how to analyze and solve problems with existing software including that for simulation. Intelligent use, however, demands the ability to evaluate and check such software with thoroughness and precision.
2. Students should learn how to critically evaluate programs written by others.
3. All courses involving calculations should make extensive use of the computer and the latest software. Application should be frequent as students progress in the curriculum. Clearly adequate computer hardware and software must be freely available to the student either through superior centralized facilities, individual PCs or both. Extensive development of professionally-written software for chemical engineering should be pursued.
4. A great deal of time can be saved in addressing designed equipment such as reactors, fractionators and absorbers by emphasizing rigorous computer calculations and the simplest shortcut procedures. Valid shortcut methods require a solid conceptual base that assists in developing an intuitive feel for a problem. However, most intermediate calculation procedures should be eliminated unless they have a real conceptual value. These methods were previously developed to provide more accuracy while still permitting hand calculation. Existing software for algebraic and differential equation solving make simulations and design calculations quite straightforward.
5. Laboratory reports should use word processing when possible, and graphical presentation of data should be emphasized, using computers with modern graphics capabilities.
6. Some hands-on experience using current practices of computer data acquisition and control with industrial-type consoles should also be encouraged.

7. In the design course in engineering, students learn the techniques of complex problem-solving and decision-making within a framework of economic analysis. The very nature of processes requires a system approach, and the ability to analyze a total system is one of the desirable attributes of chemical engineers. Rigorous economic analysis and predictive efforts should be required in all decision processes, given the availability of modern simulation tools.

8. Because of the greater diversity of interests and job opportunities today, some consideration should be given to providing a variety of short design problems representing different industries; alternatively, an opportunity can be provided for each student to select a single major design problem of greatest personal interest.

9. Faculty should be aware of the work situation of today's engineering professional. Companies have invested heavily in computers and software. Professionals are expected to use this technology, finish studies and make decisions in a much shorter time period than previously. This environment should be reflected in the educational process; yet educators must carefully develop new procedures for imparting the important values and insights that were transmitted when hand calculations were necessary. The pedagogical problem is not trivial and will require serious effort and planning to assure that students develop the fundamental habits essential in avoiding a "black box" approach to problem solving.

In a similar vein, the CACHE Corporation recently published a position paper on the expectations of the competence of chemical engineering graduates in the use of computing technology.[3] The main conclusions reached by this group identified four areas of competence. The chemical engineering graduate:

1. must be familiar with one operating system for personal and mainframe computers. This implies ability to manipulate files, perform editing and develop graphic displays.

2. must be competent in the use of at least one scientific programming language; this knowledge indicates a sufficient understanding of programming logic to test and adapt programs written by others. It is also easier to learn the second programming language once a first language has been mastered.

3. must have experience in the computer-aided acquisition and processing of information (through electronic means), as well as word processing and graphics programs for the generation of reports.

4. should gain experience in electronic mail and external databases, since this should become commonplace in the next five years.

5. should learn how to perform an evaluation and verification of programs supplied by others, since software selection is an important responsibility of today's chemical engineer.

6. should gain an appreciation of principles and concepts of numerical analysis (including convergence and stability), database management programs, spreadsheets and non-numeric programming (as used in artificial intelligence and process synthesis).

COOPERATIVE EFFORTS IN COMPUTING

It is clear that these recent studies have presented a tremendous challenge to the chemical engineering educational community, especially in the area of computing technology. Unfortunately, many existing faculty have developed course materials and teaching approaches that do not use computing. The integration of computing into these courses will not usually occur by those faculty undergoing a fundamental philosophy change. The main obstacle which precludes individual faculty from upgrading their course significantly is the lack of time and other resources. At most institutions, the introduction of computing technology takes place with the help of a few dedicated, highly involved and largely self-taught individuals; however, this has caused a tremendous disparity between universities which have managed to make their faculty and students computer literate and those who have not.

Therefore, cooperative efforts are needed to develop the necessary software and course materials, i.e., to share the resources of many departments. Software must be tested, documented and then disseminated before broad changes can occur in chemical engineering education. Over the past fifteen years, the CACHE Corporation has been the leading group in carrying out such activities.

New areas are being pursued by CACHE trustees as well as other educators, covering topics such as process design, optimization, CAI, dynamic stimulation, data acquisition and artificial intelligence. The most promising developments are discussed below.

RECENT DEVELOPMENTS

Simulation and Optimization

Once the flowsheet is specified, the solution of the appropriate steady-state process material and energy balances is referred to as flowsheeting. The essential problem in flowsheeting is to solve a large set of linear and nonlinear equations to an acceptable degree of precision, normally by an iterative procedure. While FLOWTRAN was the first flowsheeting code to be used at a large number of departments teaching process design, more comprehensive codes are now available to universities at a minimal cost. Commercial packages such such as ASPENPLUS (Aspen Technology Corp.), CHESS/CHEMCAD (Washington University and COADE Corp.), DESIGN 2000 (ChemShare) and PROCESS (Simulation Sciences) are much more advanced than FLOWTRAN and are being employed in design classes at a number of universities. Microcomputer versions of some of these programs are now becoming available. The use of these CAD tools should not necessarily be restricted to the senior capstone design course. At some schools, simulators are used in junior level courses such as fluid flow, heat transfer and separations. This fact gives credibility to the idea that engineering design can be taught throughout the curriculum. User-friendly front ends make it relatively easy for students to become familiar with utilization of the CAD software.

The combination of process flowsheeting codes with optimization algorithms is a new development which may begin to be incorporated in university teaching of process design. Optimization is an integral feature in plant design and operation. Early attempts at optimization in conjunction with a flowsheeting program used the solution of balance equations as an inner loop with the optimization subrouting choosing values of certain of the parameters in an outer loop. The optimizer usually adjusted the values of the unspecified variables or parameters and introduced revised values back into the flowsheeting program which would then perform a complete flowsheeting simulation based on those values. Revised values would be returned to the optimization routing which would again readjust the decision variables, and so on. Usually, simple search methods were used to find the optimal values of the unspecified parameters and variables so that large numbers of iterations (as many as 1,000) through the flowsheeting routines would result, at considerable cost. Thus optimization was expensive.

Optimization in the design of plants is now attractive because optimization codes suitable to solve large-scale complex problems are available, and computer and personnel costs of executing an optimization study are reasonable. Recent optimization techniques have focused on better ways of meshing the optimization procedure with the process flowsheet calculations right from the start (see Figure 1). These techniques progress toward the optimal values of the flowsheet parameters and variables while simultaneously moving towards the solution of the set of algebraic equations defining the flowsheet. Thus, they are far more efficient than older methods. Recently, commercial codes such as PROCESS and ASPENPLUS have advertised an optimization feature based on successive quadratic programming. In addition, Prof. L. T. Beigler of Carnegie-Mellon is developing an optimization feature for FLOWTRAN under the auspices of CACHE.

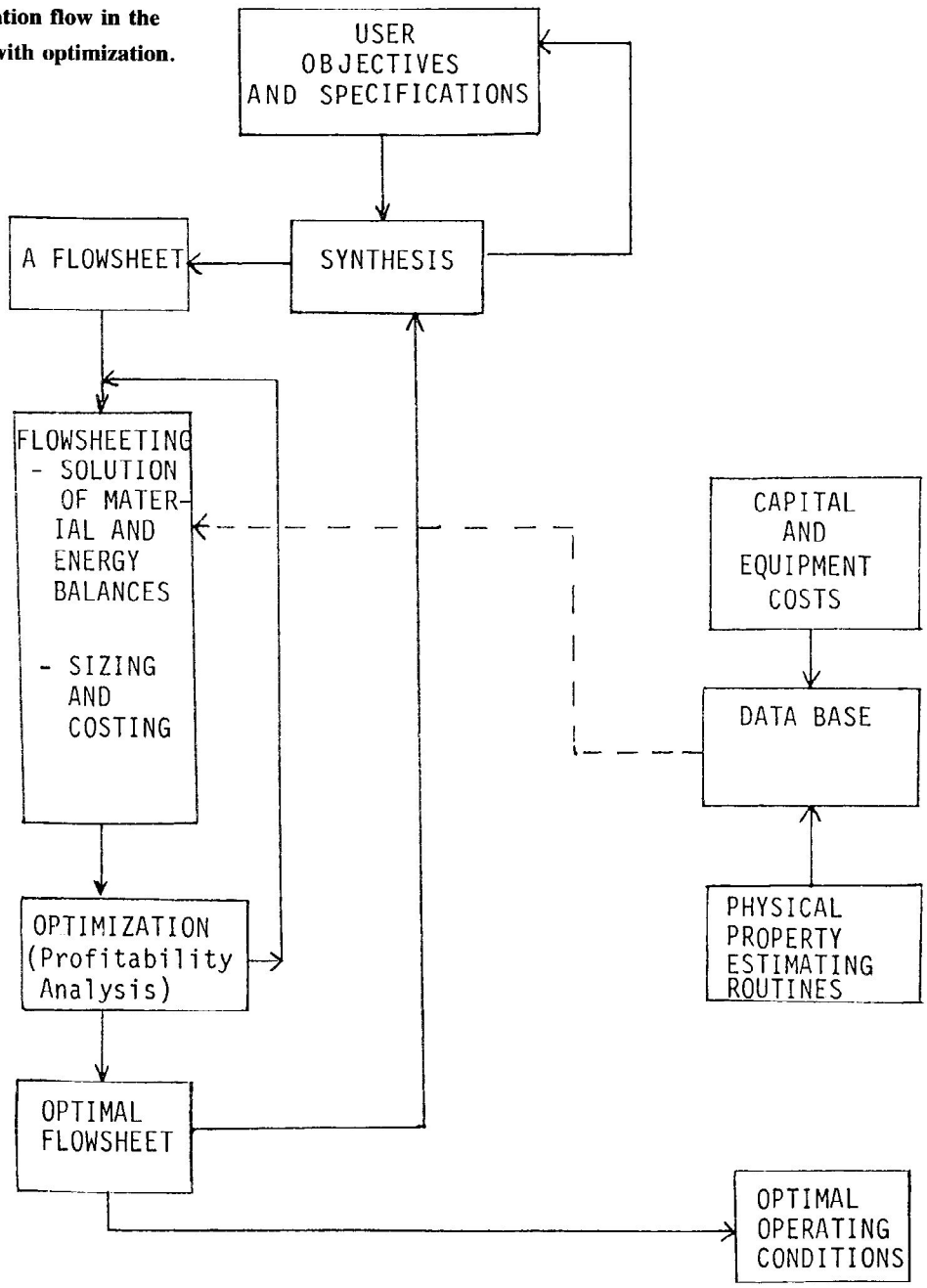
The analysis of operability of chemical plants is another topic which is slowly finding its way into the chemical engineering curriculum. A number of dynamic simulators are currently available, with perhaps the leading candidate being SPEEDUP, developed at Imperial College. This software is based on a modular subrouting structure (corresponding to physical equipment) and contains extensive physical property routines. Efficient numerical integration procedures are imbedded within the software as well, but these procedures do not demand much information from the user.

CACHE is currently negotiating to make this package available to its member departments at reduced cost, much as was done with FLOWTRAN.

Spreadsheet programs, which had been originally developed for economic analysis, are now beginning to be employed for material and energy balance calculations.[4] Information on temperatures, pressures, flow rates and chemical species classified by stream number can be listed as tables in a spreadsheet program. Simple algebraic rules can be utilized to perform calculations, with the spreadsheet providing the format. Equipment information can also be tabulated in a similar fashion. In some departments the student is introduced to such tools in the sophomore material and energy balances course.

The development of new process synthesis software is receiving increased interest. Heat exchanger synthesis and heat integration programs have been developed during the early 1980s and a few schools are testing software form Union Carbide (ADVENT) and Linnhoff-March (TARGET II). CACHE is evaluating these packages and is currently distributing TARGET II to universities. In addition, J. M. Douglas of the University of Massachusetts is developing two new packages: PIP, a process invention procedure, and GIGAS, a distillation synthesis program.

Fig. 1: Information flow in the design process with optimization.



Computer-Aided Instruction

Computer-aided instruction (CAI) has proven to be a valuable technique for teaching chemical engineering principles. In CAI systems, the student is essentially led through a series of exercise problems which are solved at the terminal. The student enters a proposed solution to the problem directly or by selecting one of a multiple choice of responses. The student's response is evaluated by the program, and, depending upon the result, the student is advanced to the next problem category or is presented with a suitable remedial problem. Typically, such systems also include capabilities for the collection of student performance and usage data and for aggregating problem effectiveness data for instructor feedback. These packages, can, however, reduce the need for recitation sessions and the submission and manual grading of regular exercise problems. Consequently, they can reduce the demand for teaching assistants and recitation instructors in the core ChE courses. Examples of CAI systems include PLATO and MicroCACHE.

Because of the recent emphasis on open-ended design problems in the chemical engineering curriculum, CACHE has been developing prototype computer-based lessons in core subjects where computer use has historically been inadequate. However, subjects such as process design and control are not being emphasized since those subjects have a large number of computer-based tools already available. One other goal in this project is to expose the undergraduate student to problems in nontraditional areas of chemical engineering, such as electronic materials processing, biochemical engineering, and the food industry. Lessons will be provided on a floppy disk compatible with the IBM PC. They must be self-contained, easy to install, and understandable to a faculty member inexperienced with computers.

Five prototype modules are being prepared currently. One deals with supercritical fluid extraction. The program introduces design concepts to them about supercritical fluid extraction, a relatively new separation process. The lesson is divided into three parts: the first two sections introduce the student to the theory of supercritical extraction and explain how to design the process; the third part consists of a menu-driven program that allows the student to design a simplified supercritical extraction process. With the easy-to-use menus and on-screen instructions, students may easily vary any of the design parameters such as flowrates, compositions, pressure, temperature and solvent-to-feed ratio.

Another lesson concerns the design of a unit for the removal of SO_2 from an air stream by scrubbing with water. The unit is a counter-current gas absorption tower in which the gas to be cleaned is fed to the bottom of the unit and counter-currently contracts a falling aqueous liquid in the column. In the problem, the student is led through the basic design equations that need to be solved to predict the length of the absorber for specified gas and liquid flow rates and degree of removal. The problem, which is intended for use in a senior course on separation processes, teaches the student about the effect of chemical equilibrium on gas absorption and the numerical techniques needed to obtain a solution. The student may examine the effect of different specifications on the ultimate length of the column required.

Some CAI programs are now available to reinforce textbook material. In conjunction with the book ***The Elements of Chemical Reaction Engineering,*** H. S. Fogler of the University of Michigan has developed nine interactive programs which involve various reactor design calculations as well as simulation, troubleshooting and game playing. The reason so few books currently include computer-aided instruction materials is the large amount of time required to develop such lessons, e.g., 200 to 1,000 hours per lesson by a faculty member plus a programming assistant. Probably the most fertile area for textbook software for personal computers is in numerical analysis, and by next year there should be a good selection of such materials. However, most university mainframe computers contain IMSL or other packages, providing access to much of the standard numerical analysis software.

Process Monitoring and Control

The increased use of computers for process control has caused some changes in how process control is taught to undergraduates. First it is clear that in most industrial operations, computer control systems are used as opposed to analog control systems. The digital control systems of the 1970s based on minicomputers have given way to microcomputer-based systems; for some cases personal computers such as the Apple II+ and the IBM PC series

have become quite attractive for data acquisition and control.[5] With standard interfaces available, such digital systems are relatively easy to implement and fairly inexpensive. A recent survey of North American chemical engineering departments published by CACHE verifies the widespread use of digital computers for experimental data acquisition and control. The increased standardization of such equipment has reduced the need to teach specialized courses in real-time computing in chemical engineering departments, as was necessary in the 1970s.

There are a few universities where commercial control systems are being used to expose students to industrial control interfaces, e.g., Honeywell TDC 3000, Fisher Controls PROVOX and IBM ACS. Several universities including LSU, Purdue, Waterloo and Imperial College, have installed the ACS software on an IBM 4341 machine. Koppel and Sullivan[6] have carried out extensive efforts to develop educational software so that ACS could profitably be used as an educational tool. The initial work focused on the dynamic simulation of a furnace which is imbedded in the computer, rather than control of an actual process. G. R. Sullivan at Waterloo has combined the dynamic simulator SPEEDUP with ACS to provide a more comprehensive tool called SPEEDWACS.

At this time, no dynamic simulator is universally used in most undergraduate control courses. In fact, there has been a general lack of good, proven, professional-quality software for use in the teaching of process control until recently. Some promising packages ought to be mentioned: one is program CC which is a linear control system design package based on single input/single output systems developed at the Department of Electrical Engineering at Cal Tech. CC runs on the IBM PC and is available from Systems Technology, Inc. of Hawthorne, California. It contains both time and frequency domain analyses. A recently developed mainframe program is CONSYD,[7] developed at the University of Wisconsin and now under joint development with Cal Tech. CONSYD is a more comprehensive program than CC and treats multiple input/multiple output systems. CONSYD includes modern control algorithms as well as the classical approach used by CC.

Even with the availability of digital control systems and software, the subject matter taught in undergraduate process control courses has not changed substantially in the past ten years. An emphasis on design of PID controllers still remains, although the use of such tools as frequency response to optimize performance is now feasible because of interactive computer graphics.[8] The teaching of sampling theory, z-transforms and discrete-time control still remains vested in the graduate level curriculum; however, advanced control techniques that require digital implementation such as feedforward control, time-delay compensation and adaptive control can still be taught using continuous time analysis.

Artificial Intelligence

The use of artificial intelligence in the undergraduate curriculum could become fairly extensive when one considers many possible applications of AI:

1. process design support (mechanical design, instrumentaion specifications, catalyst selection),
2. design of molecules with desired physical/chemical properties,
3. synthesis of process flowsheets and/or novel processing schemes,
4. front-end of a sophisticated design simulators,
5. discrete event planning and analysis (e.g., batch processing),
6. troubleshooting and alarm analysis, and
7. analytical mathematical calculations (e.g., MACSYMA).

There are a variety of junior/senior courses where AI software could be used; there may also be an occasional course on AI taught in chemical engineering at the undergraduate level, although this is more likely done at the graduate level.

Venkatasubramanian[9] recently described a graduate level course in chemical engineering on knowledge-based expert systems. The course covers knowledge representation, searching procedures (forward/backward chaining), knowledge acquisition, expert system development, symbolic computational methods, qualitive reasoning, process plant diagnosis and expert systems tools and shells for process engineering. The students develop rule-based expert systems during the course. Examples of the projects include troubleshooting of chemical plants, a design consultant for plastics selection and design, expert systems for pump selection

and agitator selection, and physical property estimation.

As discussed by Stephanopoulos,[10] the application of AI or knowledge-based expert systems will use different programming languages and hardware than is typically provided in undergraduate programs. Several versions of LISP are used for expert system development, along with special purpose languages like PROLOG or Smalltalk. Machines manufactured by LMI, Symbolics, Xerox and Texas Instruments are especially well-suited for development of AI software, but equipment from DEC, SUN and even IBM PCs can be employed with various dialects of LISP. It is expected that expert systems software will become more readily available during the next five years. Prof. G. Stephanopoulos of MIT is leading the new CACHE task force to help integrate AI into the curriculum.

CONCLUSIONS

The integration of computing into all courses in chemical engineering is a generally accepted goal among educators; unfortunately, there are many obstacles to realizing this goal.

Because computing technology continues to grow at a rapid pace, it is doubtful that most academic departments will be able to provide the latest software and hardware for its students. However, cooperative efforts in software development and dissemination such as those described in this paper may ameliorate the software development program. The hardware problem will only be solved with significant government and industrial help.

REFERENCES

1 Denn, M. M., "Design, Accreditation, and Computing Technology," ***Chem. Engr. Educ.***, XX, 18 (1986).

2 Septenary Committee, "Chemical Engineering Education for the Future," University of Texas, Austin, TX (1985). Also ***Chem. Engr. Prog.***, 81, 9 (October 1985).

3 CACHE Corp., "Expectations of the Competence of ChE Graduates in the Use of Computing," ***Chem. Engr. Educ.***, XX, 19 (1986).

4 Grulke, E. A., "Using Spreadsheets for Teaching Design," ***Chem. Engr. Educ.***, XX, 128 (1986).

5 Joseph, B. and D. Elliott, "A Microcomputer-based Laboratory for the Teaching of Process Control," ***Chem. Engr. Educ.***, XVIII, 136 (1984).

6 Koppel, L. B. and G. R. Sullivan, "Using IBM's Advanced Control System," ***Chem. Engr. Educ.***, XX, 70 (1986).

7 Morari, M. and W. H. Ray, "CONSYD-Control System Design," ***Comp. Chem. Engr.***, 10, in press (1986).

8 Heeb, R., T. F. Edgar and J. O. Hougen, "Computer-Aided Process Control System Design Using Interactive Graphics," ***Comp. Chem. Engr.***, 5, 225 (1981).

9 Venkatasubramanian, V., "A Course in Artificial Intelligence in Process Engineering," ***Chem. Engr. Educ.***, XX 188 (1986).

10 Stepanopoulos, G., "A Research Program on Artificial Intelligence in Process Engineering, " ***Chem. Engr. Educ.***, XX, 182 (1986).

This article is reprinted from the I.CHEM.E. Symposium Series No. 101, with permission.

INTERHEAT/HEATNET

contributed by Prof. Per A. Loeken
of INTERSOFT
Trondheim, Norway

The INTERHEAT program consists of data input, storage/retrieval, composite curves, advanced targeting, network design and network optimization. It is highly interactive with a user-adaptable interface, including menu jumping, command processing and tutorial facilities to enable new users easy access. This is ideal for educational use within universities.

The program may be made available in the form of EXE files for VAC computers running under the VMS operating system. The program system has an EXE file of approximately 2,100 blocks and the total disk space required is approximately 4,400 blocks including some data files.

The program can run on different terminals, specified by the user, including standard ANSI terminals giving alpha-numeric graphical diagrams. By using graphical terminals as VT241/240 or Tektronix 4105, the results are presented in professional color graphics form. The figures represent sample diagrams.

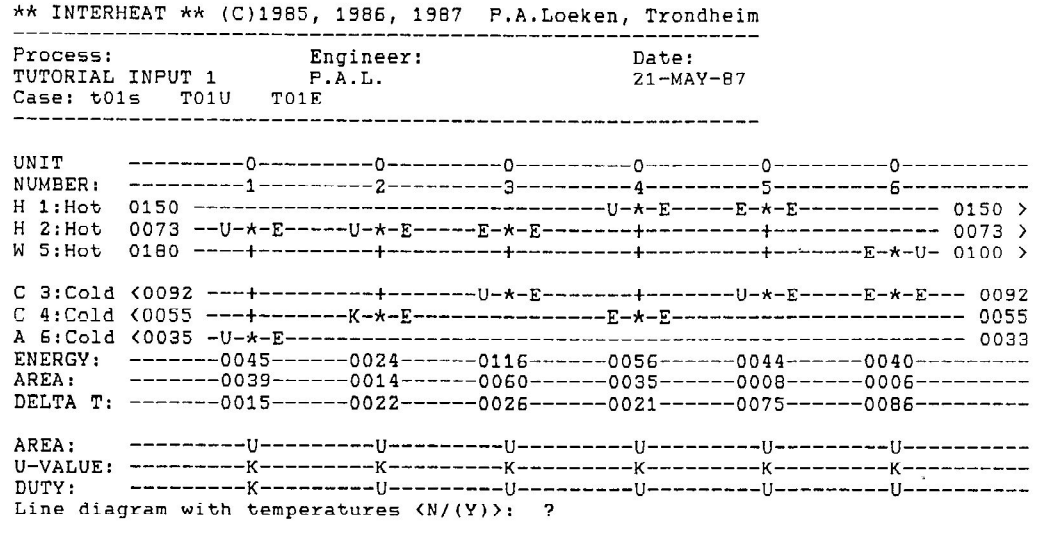
An IBM mainframe version (VM/CMS and ACS system) and a Prime version are under development.

A separate agreement would have to be arranged with the individual university, including security clauses. The use of INTERHEAT would have to be limited to students and university personnel only. Usage by these groups may be unlimited. For any consultancy activity in energy analysis to industrial customers, a separate agreement may be arranged.

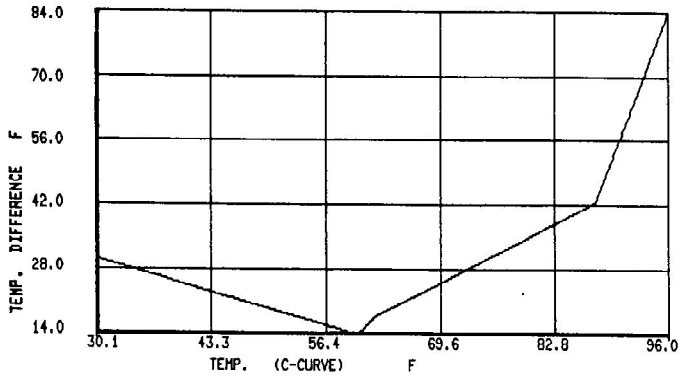
A university license fee of $400 per installation would cover this program, supported by a user and tutorial guide including a relevant case.

A cooperation with the National Engineering Laboratory, U.K. (NEL) is established. Together we have linked INTERHEAT, PPDS and some other programs into one package named HEATNET.

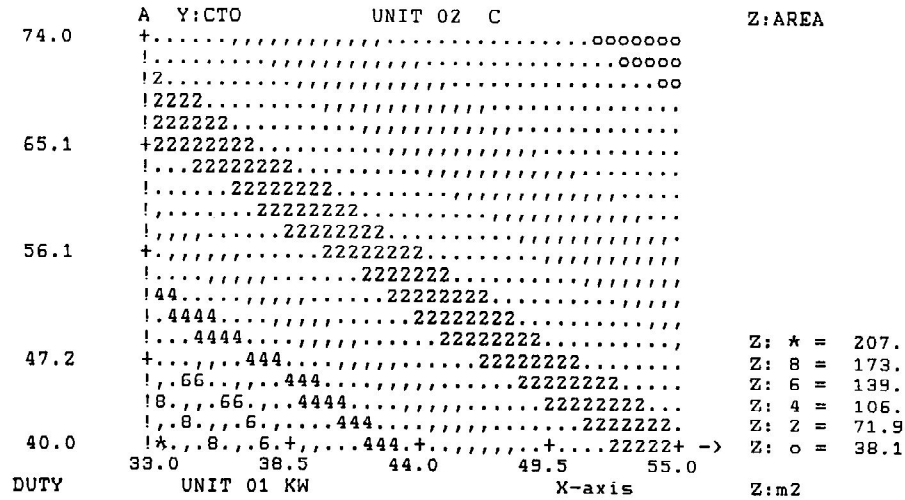
The inclusion of a physical data system is normally of interest to users. One possibility is to use NEL40, a physical property package with forty components to be linked as a part of HEATNET. Because this is owned by NEL, its availability must be discussed with them.



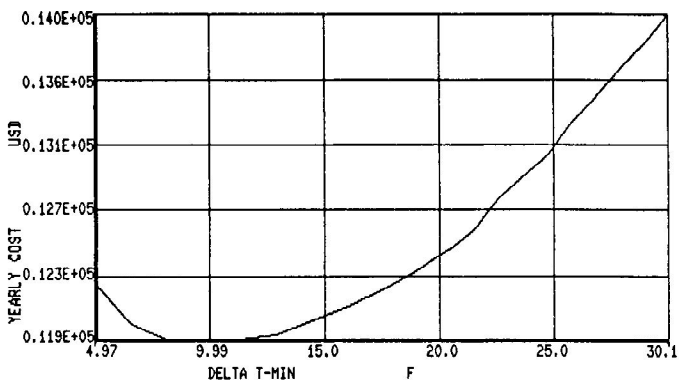
Process line diagram for a designed process.



Example of a composite curve.



Example of contour diagram showing pay-back time as a function of two selected variables.



Example of an advanced targeting curve.

BITNET, A COOPERATIVE NETWORK

by Maurizio Rovaglio
Dipartimento di Chimica Industriale
ed Ingegneria Chimica "G.Natta"
Politecnico di Milano (Italy)

Present address:
Department of Chemical Engineering
University of Massachusetts
Amherst, Massachusetts 01003

BITNET (Because It is Time NETwork) is a cooperative network serving over 1800 hosts located at several hundred sites (mostly universities) in 26 countries. The table summarizes the distribution of the sites and shows how, collectively, they constitute a worldwide "meta-network."

BITNET is a communications link between universities and research centers with few requirements or restrictions other than that each site must acquire a leased line to facilitate the connection to another BITNET node and, in the spirit of a cooperative network, be willing to serve as a connection node for at least one new member.

BITNET had its beginnings in 1981 when the first two nodes, City University of New York (CUNY) and Yale University, were connected on May 5. Since that day the coordinated efforts of the first two sites and the support of IBM gave rise to a very rapid development of the communications link among different universities in the United States and, later, all over the world.

BITNET started as a network of IBM hosts. Today, although most BITNET hosts still communicate using an IBM communication environment, there are also many non-IBM hosts that use emulation software to provide the appropriate protocols for DEC, VMS, UNIX and Sperry environments. Hosts are interconnected by leased phone lines supporting 9600 bps data transmission.

There are three main constituents of the network: BITNET in the United States and Mexico, NETNORTH in Canada, and EARNET in Europe. There is also ASIANET in Japan, Taiwan, and Singapore, and there are plans to extend the network to South America. The distinctions among the different "clusters" are purely political; mail, files and other types of communications can be freely exchanged between any two hosts.

NETNORTH provides communications for a great number of Canadian academic and research sites, and was designed using the same technology and several of the same basic assumptions as BITNET. The network currently consists of over 100 nodes; direct links exist to BITNET and EARNET.

EARNET (European Academic Research NETwork) links over 500 hosts and over 100 institutions in 20 countries. EARNET is very similar to the previous two "clusters" and it is based on the same design principles and philosophy. EARNET is an integral part of BITNET, which means that all the European nodes are listed at all U.S. BITNET sites, and vice versa. The communications link between the two continents is by satellite, and the data transmission speed is usually 64,000 bps.

BITNET was originally used primarily for collaboration and communications among systems programmers at university computation centers. At present, the entire network is used by scholars, administrators, faculty, staff and students from a variety of different disciplines. Services provided include electronic mail, file transfer and interactive messages. The interactive messages allow several users to communicate interactively with only a moderate delay, usually less than eight seconds.

With these kinds of "tools" it is really possible today to work without limits or boundaries and to keep closely in touch with different research groups, exchanging data, information, software or whatever is necessary for a common research program.

In addition to summarizing the spirit of BITNET, this short note demonstrates the prac-

tical and powerful applications of the network when a fast and easy contact line is needed between two universities. On this subject, let me give some details about my personal experience.

It was very convenient and easy for me, working with BITNET, to move programs and data banks from my university, Politecnico di Milano, in Italy (IMIPOLI) to the University of Massachusetts (UMASSVM), where I am currently working with Professor M. F. Doherty. It is also very helpful for keeping me in touch with my original advisors, listening to all the news about my Ph.D. program, and exchanging interesting information and reflections with my colleagues in Italy.

As to the speed and reliability of the communications link, my experience has been that mail can be exchanged usually in a couple of hours and that interactive messages require a very small time delay (10 seconds on average). Even the file transfers are very fast and extremely reliable. For example, we transferred about 30,000 lines of programs and data banks in approximately 5 hours without any transmission error.

An unusual feature of BITNET is that there is usually exactly one path between any two hosts. This means that if one of the nodes on the route is down or disconnected, it becomes impossible to communicate between the desired hosts. This defines a limit to the network's reliability. An automatic multiple routing is hoped for in the near future.

In summary, let me conclude that, from the user's point of view, BITNET represents a powerful communications channel and a possible door opened on the world.

BITNET Sites List as of July 6, 1987

Countries	# of Sites
BITNET:	
United States	314
Mexico	1
NETNORTH:	
Canada	52
EARNET:	
Austria	8
Belgium	15
Denmark	13
Finland	9
France	64
Germany	112
Great Britain	1
Greece	3
Iceland	1
Ireland	2
Italy	45
Israel	19
Norway	4
Netherlands	33
Portugal	2
Spain	9
Sweden	12
Switzerland	20
Turkey	7
ASIANET:	
Japan	15
Taiwan	1
Singapore	1

MICROCOMPUTER SOFTWARE REVIEW

Comments on Mathwriter for the Macintosh and an example of efficient use of microcomputers by students.

by Bruce A. Finlayson
University of Washington

Review Mathwriter, an equation-writing software for the Macintosh

There are few computer programs that revolutionize how I work, but Mathwriter has. The program was designed for a technical writer and I find it easy to use.

The equations appear on the screen as they will in the document, making for easy editing. The program can of course do all the standard symbols: integrals, summations, fractions, matrices. As an illustration of the excellent design, when an integral is chosen, the computer immediately moves to the location for the lower limit of integration and automatically changes the font size to a smaller size. Superscripts and subscripts can be chosen automatically with the mouse and the font size is automatically reduced.

Greek letters are displayed at the bottom and can be chosen with the mouse. This is easier than changing the font back and forth manually. There are many options available; pallets containing all sorts of mathematical symbols can be added to the bottom as well. Equations, once formed, can be edited, copied and pasted into MacWrite and Word.

The program is especially useful for someone using lots of indices. You can create macros like

$$c_i^{n+1}.$$

Each time that symbol is needed it can be recovered with one key stroke. I found this feature especially useful when constructing finite difference formulations for transient problems. The ability to edit equations makes it almost faster to do the equations yourself than have your secretary do them. You save time when an equation is modified for use in different parts of the document. I am able to do the modifications directly on the screen and avoid ever writing down revisions on paper. The clear copy from the Laserwriter makes for a satisfying result. The program creates the Postscript file needed by the Laserwriter. A version is also available to create a TEK file.

There is only one drawback. Once an equation is pasted into MacWrite, you cannot edit it. This means that you save a Mathwriter file containing the equations and also paste them into your document. If you need to revise an equation you start with the Mathwriter file. This is because the Mathwriter format includes many more options than are needed to print the result, and only the needed ones are saved in the transfer to MacWrite.

The equations documents are large, but the MacWrite documents are not. When there are lots of super- and subscripts the transfer to MacWrite is slow on a Macintosh 512K, but fast enough on a MacPlus. Once I learned to operate in this way there were no particular problems.

I have only tested this program with MacWrite, although it is available for other word processing packages.

This software has affected the way I work. I can create equations first on the computer—my handwritten notes are brief since they only include the information necessary to recognize the equation. Algebraic rearrangement can be done easily. The result is text that is clearer with simpler steps. I have used the program extensively when writing a book and it is a virtual necessity now.

The Mathwriter software is available for $49.95 from Cooke Publications, P.O. Box 4448, Ithaca, New York 14852.

A Case Study Where the Microcomputer Made a Real Difference

During Spring quarter 1987, I was teaching a course entitled "Nonlinear Analysis in Chemical Engineering." I had discussed the application of the finite difference method and the finite element method to the convective diffusion equation,

$$\frac{\partial x}{\partial t} + Pe\frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}$$

but had not given detailed stability results. For the finite difference equation we did derive the stable step size for the diffusion equation and an explicit method of integration. We did not derive the corresponding stability criterion for the Galerkin finite element method; indeed, it is rather difficult to derive because the derivation involves complex algebra and the use of complicated trigonometric identities. The result is not often stated in books, either. To study the stability question for both the Galerkin finite element method applied to the diffusion equation and both methods applied to the convective diffusion equation, I provided a computer software package, CDEQN, which integrated those equations and plotted the results. (An article on CDEQN follows.) The program was in compiled BASIC on the Macintosh and utilized pull-down menus, buttons, etc., so that it was easy to use.

The following problem was assigned:

Solve the convective diffusion equation in the following cases. The software package CDEQN is available to run on the Macintosh with graphics.

a) Set Pe = 0 to study the diffusion equation. Use Δx =0.2 (for x from 0 to 1) and D = 1. Integrate to t = 0.1 using the Euler method. First choose a step size based on the stability criterion. Do one step and verify the computer calculations. Then do several steps to reach t = 0.1. Resolve the problem using time steps that are larger than that required for stability in order to see what happens when the stability limit is disobeyed. Compare the finite difference method with the Galerkin finite element method. Discuss your results.

b) Set Pe = 100. Pose one question to investigate. Typical questions could be:

   (i) What are the Δt requirements of the different methods?

   (ii) How does the GFEM solution change as Δx is decreased?

   (iii) How does the needed Δt change for (ii)?

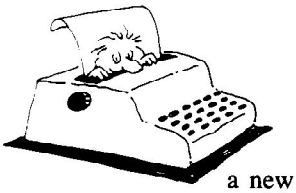
   (iv) Compare solutions with different upstream weighting parameters.

What excited me was the results. Everyone in the class honed in on the Galerkin stability limit of $\Delta t \leq \Delta x^2/6$ for for the diffusion problem, or a Δt three times smaller than for the finite difference method. The students deduced this result in an experimental fashion. The program ran quickly enough that they would pick a Δt , run the problem, see if the method was stable by looking at the graph and seeing if the oscillations grew with time. If so, they would reduce the step size. This is much quicker than going through the von Neumann analysis to deduce the result. So in this phase of the problem they got results fifteen to twenty minutes faster than any technique other than if I gave them the answer.

The next part to the assignment (for a Peclet number of 100) also led to revealing answers. Many in class came up with a stability limit involving $\Delta t / \Delta x$, i.e., a Courant number limit, despite the fact that we had not talked about this type of limit in class at all. Again their approach was experimental—choose Δx , vary Δt , until the results are stable, using the graph as the criterion, and repeat for a different Δx. One student even observed a phenomenon, explaining that the von Neumann analysis of stability and the matrix norm say the method is stable—the disturbance will eventually die out—but the von Neumann analysis says it is unstable, that the disturbance keeps growing. The resolution to this paradox is that the disturbance grows as the front moves to the right, but the disturbance dies out behind the front and approaches the correct long time solution. Both methods are correct in what they predict. There are several research papers discussing this topic, and one student observed the phenomenon by "playing" with the program.

I have used several microcomputer programs in a variety of classroom situations. Most of the programs do what can already be done on other computers, perhaps easier, with perhaps easier access. This experience, however, gave a qualitatively different result: the students obtained a whole new dimension to their education by using the program.

MICROCOMPUTER CHEMICAL ENGINEERING PROGRAMS



a new column edited by Bruce A. Finlayson, University of Washington.

Have you wondered what microcomputer programs are being used in other chemical engineering curricula? A new feature has been added to the CACHE newsletter. Bruce Finlayson will edit a column as titled above that provides a mechanism for university professors to let others know about the programs they have developed and are willing to share on some basis.

There are two categories: a) completed programs, and b) programs for testing. The ***completed programs*** can be described by a 250-word description of the program, machine requirements and ordering information. These programs should be ready to be shipped out the door and should have been tested by students at more than one university. The ***programs for testing*** are listed with title and one to two sentences. These are programs the author would like tested at other universities. It would be helpful in both cases if the specific chemical engineering course were identified in which the program is useful. The programs will not be reviewed by Prof. Finlayson, nor will they be certified by CACHE.

In order to edit the column efficiently, submissions must be made to Finlayson via BITNET using userid 27432 and node UWAV4. He will acknowledge receipt of the submission via BITNET and will send the edited column to the CACHE office via BITNET. Letters will not be accepted or acknowledged. This requirement has two goals: to reduce the need for secretarial typing and to encourage academic chemical engineers to use electronic mail. Since anyone writing a computer code is computer literate, they can figure out how to use BITNET at their local installation when the incentive is exposure for their program. They can then share the protocols with their colleagues for other uses. Let us hear from you!

Here are the first announcements:

Convective Diffusion Equation (CDEQN)

by Bruce A. Finlayson, University of Washington

Program CDEQN solves a partial differential equation using either the finite difference method or the Galerkin finite method. The equation solved in CDEQN is the convective diffusion equation:

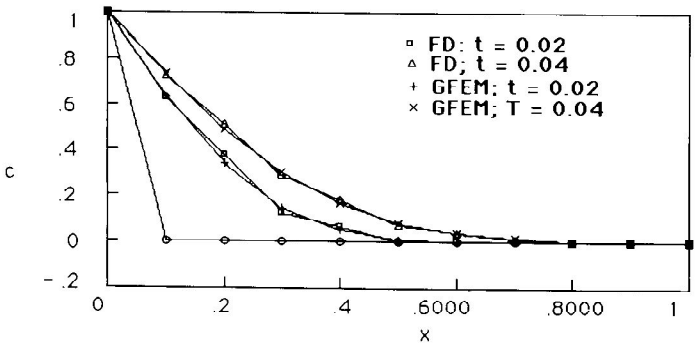
$$\frac{\partial x}{\partial t}+Pe\frac{\partial c}{\partial x}=\frac{\partial^2 c}{\partial x_2},\quad 0\leq x\leq x_L\leq 2.$$

The initial conditions are c(x,0) = f(x), where the function is set using the initial condition options in the program. The boundary conditions are

$$c\,(0,t\,)=c_0,\quad \frac{\partial c}{\partial x}(x_L\,,t\,)=0.$$

The diffusion term can be left out to obtain the advection equation and the convection term can be left out to obtain the diffusion equation. The integration in time is with a first-order Euler method. Options include solving for the spatial dependence with either the Galerkin finite element or finite difference method. Upstream weighting can be included or not as you wish. The results are plotted in the program ENGNPLOT (included). It runs on a Macintosh using pull-down menus. A sample result is shown.

This program has been used in a numerical analysis course, but is suitable for undergraduate heat and mass transfer courses. It runs on the Macintosh with 512K memory. The diskette with the system and the program on it requires an 800K drive, but the program itself fits on a 400K drive. A disk containing this program may be purchased for $10, payable to Bruce A. Finlayson. Send requests to: Bruce A. Finlayson, 6315 22nd Avenue NE, Seattle, Washington 98115. The compiled program can be copied.

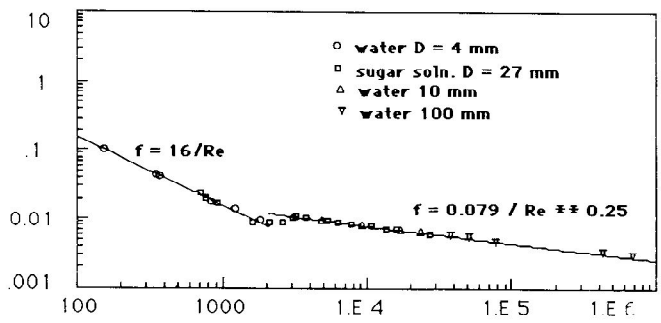


Diffusion Equation Solved with GFEM and FD Method

Engineering Plot (ENGNPLOT)

by Bruce A. Finlayson, University of Washington

The program ENGNPLOT is designed to make plots easily on command of the user. The program responds to either pull-down menus or typed commands. Basically the user enters the data to be plotted and the plot is made. The form of interpolation is linear between data



Friction Factor-Reynolds Number for Flow in a Pipe

points. The user can control the scale (log or linear), the range of the graph, the number of subdivisions, the titles, the number of lines drawn, the use of symbols and connecting lines. Up to twelve captions can be placed anywhere on the graph. These captions can use several fonts, boldface, underlines, etc. The data can be read from a file or entered from the keyboard. The plots can be printed or saved as a screen image; this can then be put into MacPaint, and hence into any Macintosh document. A sample plot is shown.

The program can be used in a stand-alone fashion in courses such as laboratory. It is used as the plotting package for other programs mentioned in this issue and has been thus used by many students in different courses.

The program runs on the Macintosh with 512K memory. The diskette with the system and the program on it requires an 800K drive, but the program itself fits on a 400K drive. A disk containing this program may be purchased for $10, payable to Bruce A. Finlayson. Send requests to: Bruce A. Finlayson, 6315 22nd Avenue NE, Seattle, Washington 98115. The compiled program can be copied.

Short Cut Distillation and Flash Calculations

by Bruce A. Finlayson, University of Washington

This program is designed to run in an interactive mode. It has the capability of doing both flash calculations and short cut distillation calculations, both with multicomponent systems. The thermodynamics package is general, with the vapor phase fugacity calculated by the Redlich-Kwong equations of state. The liquid phase is considered a non-ideal solution with activity coefficients based upon the Hildebrand solubility parameter. The short cut distillation module uses the Fenske total reflux equation, Underwood's equation for minimum reflux, and Gilliland's correlation for the number of stages. Thermodynamic parameters are included for 98 different chemicals.

The short cut distillation package requires the user to specify the ratio of reflux rate to minimum reflux rate, pressure of the column and fraction of the light and heavy key. The output from the programs gives composition and physical properties (including K values) for the products, number of stages, feed stage and condenser and reboiler duties. The column is sized and the capital and operating costs are calculated. The user can change the economic parameter, if desired. The equivalent uniform annual cost (EUAC) is also calculated.

This program is a version of the same program that has been used in design applications for over ten years. It has been ported to both the Macintosh and the IBM PC. It uses no graphics. On the Macintosh it runs on a 512K machine with a 400K drive. On the IBM PC it runs on a machine with 256 KB memory. A disk containing this program and documentation may be purchased for $10, payable to the University of Washington. Please specify which machine you plan to use. Send requests to the Department of Chemical Engineering, BF-10,

University of Washington, Seattle, Washington 98185 to the attention of Bruce Finlayson. The compiled program can be copied. The IBM version will also be distributed at some future date by CACHE through their Curriculum Task Force.

Vapor Compression Refrigeration Cycle & Compression of an Ideal Gas

by Stanley Sandler, University of Delaware

This disk, for an IBM PC or compatible, contains two tutorial used in a first course in chemical or mechanical engineering thermodynamics. The first problem, RANKINE, begins with a short review of a vapor compression refrigeration cycle. The student is asked to trace out the cycle on a pressure/enthalpy diagram for refrigerant-12 and then read the properties at each state from this diagram. Then by using the data collected and the mass, energy and entropy balances, the student computes the rates which work needs to be supplied to the compressor and heat absorbed in the evaporator. Finally, the coefficient of performance is computed.

The second lesson, GAS, deals with the expansion of an ideal gas into cylinders which are partially or totally evacuated. This problem uses only the mass and energy balances, and the ideal gas equation of state.

To run these programs, an IBM compatible with DOS 2.0 or higher, 256KB of memory and one disk drive is needed. A disk containing these programs may be purchased for $29 from Customer Services, Office of Instructional Technology, University of Delaware, Newark, Delaware 19716.

Computer-Aided Analysis for Process Systems

by Ted Cadman, University of Maryland

This is a series of PC-compatible programs for material and energy balances of chemical processes. The series is intended as the primary computational tool in the introductory course in chemical engineering. The series automates routine bookkeeping, organizational and computational tasks of analysis. The logical steps of analysis have purposely not been automated—rather a user-friendly environment is provided. A user's manual contains twenty-five chapters illustrating the use of the series with problems of increasing complexity.

The capabilities of the programs include twenty streams and ten systems, including recycle. The material and energy balances use tools appropriate to the sophomore level—ideal mixing. The student can compute dew and bubble points, adiabatic flame temperatures, heats of reaction and do distillation. There is an automated degree-of-freedom analysis. The student can insert known data and then ask the program to do a degree-of-freedom analysis about a system. The computer then indicates which variables have been specified and which ones still need to be specified before the problem can be solved.

The programs come on two diskettes for the IBM PC. They work with high resolution graphics (the University of Maryland uses the IBM color graphics card). The code is compiled MS BASIC, so no licenses are necessary. They have been used in class for three years. To obtain the diskettes, send a check for $20, payable to the Engineering Research Center, to Ted Cadman, Chemical Engineering, Engineering Research Center, University of Maryland, Baltimore County, Catonville, Maryland 21228.

EURECHA Teaching Project

by Dr. Z. Fonyo, ETH-Zentrum

The EURECHA Teaching Program Project collects and distributes chemical engineering programs to teaching institutions. The modifications make the programs suitable for almost all machines with FORTRAN compilers and can be used interactively or in batch mode. Some programs can be run on a personal computer.

CHEMCOSET	Data bank.
UNICORN	Flowsheet programs.
DISTILSET	Distillation programs.
THERMDINSET	Thermodynamics
CAPCOS	Capital cost estimation package.
TACS	Simulation of control systems.
VLESET	Comprehensive VLE analysis package.
DIAGNOSE	Chemical plant fault diagnosis exercises.
SYNSET	Heat exchanger network and column sequencing.

STATCHAR	Statistical characterization of analytical results.
BATCHDIST	Batch distillation design program.
INTERN	The designs of column internals.
REACTEX	Eleven reactor models suitable for running alone or as UNICORN subroutines. All have been used as the basis for student design projects.
REACTPELL	A heterogeneous solid/gas catalytic model of the partial oxidation of benzene to maleic anhydride.
ENISYN	Synthesis of energy integrated distillation systems.
OPTIMISER	Nonlinear optimization in chemical engineering.
REPROCHE	Nonlinear regression program.
CHEQUUS	Chemical equilibrium calculation.

The price for one source code is 300-400 Swiss francs. For further information, contact:

Dr. Z. Fonyo
ETH—Zentrum
Technisch-Chemisches Labor
CH-8092 Zurich, Switzerland

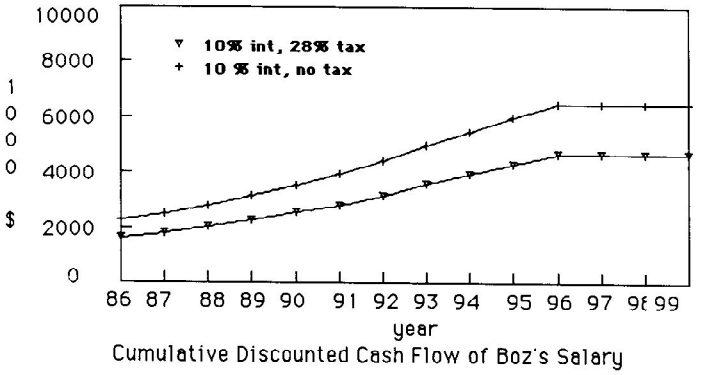
Discounted Cash Flow Analysis (and Present Worth)

by Bruce A. Finlayson, University of Washington

Program PROFIT does a discounted cash flow analysis of an investment. The information such as capital investment, revenue or expenses, can be different each year or be the same from year to year. The user must define the investments by specifying the capital investment, when it is made, when the service life begins and ends, what the working capital is and the salvage value. The revenue and expenses are also specified year-by-year. The tax rate is specified and the method of depreciation is either straight-line, double declining balance or sum-of-the-years'-digits. The program then calculates the discounted cash flow rate of return. The present worth for a given interest rate can also be calculated.

The capital investment, revenue and expenses, and discounted cumulative cash flow (for various interest rates) can be plotted using the program ENGNPLOT (included). The program runs on the Macintosh using pull-down menus. A sample output is shown; this is the discounted cash flow of the $11 million football contract of rookie Brian Bosworth of the Seattle Seahawks.

This program has been used in the design course and runs on the Macintosh with 512K memory. The diskette with the system and the program on it requires an 800K drive, but the program itself fits on a 400K drive. A disk containing this program and documentation may be purchased for $10, payable to the University of Washington. Send requests to Department of Chemical Engineering, BF-10, University of Washington, Seattle, Washington 98195, Attn: Bruce Finlayson. The compiled program can be copied.



Cumulative Discounted Cash Flow of Boz's Salary

IMPORTANT NOTICE

TO: Faculty in Chemical Engineering

FROM: Peter R. Rony, Joe Wright, and Norman Rawson
CACHE Task Force on Electronic Mail

SUBJECT: BITNET User Identification Numbers

The testing of an experimental IBM software package, called GRAND, as a world-wide wide-area network file server for chemical engineering departments will commence by the end of 1987. It will be located on a host IBM mainframe within the chemical engineering department at Louisiana State University.

We request that you provide at least one userid and node address, but preferably all userids and node addresses, associated with your department.

Please send this information to Peter Rony (Department of Chemical Engineering, Virginia Tech, Blacksburg, VA 24061, RONY AT VTVM1). We will compile the first comprehensive listing of BITNET userids in chemical engineering and mail it to CACHE member departments late in the fall. Thank you for your assistance.

The current CACHE listing of BITNET userids is given in the table. Observe that the most useful userids contain the last name (and possibly initials) or just the initials of the faculty member. For example, a userid such as RONY, PRONY, PRRONY, or initials PRR (least desirable) is preferred for the CACHE electronic mail system.

BITNET USER IDENTIFICATION NUMBERS

ID	NAME
LOT AT PSUVM	Richard LaRoche, Penn State
SCEF0003 AT WSUVM1	Jim Petersen, Washington State University
RONY AT VTVM1	Peter Rony, Virginia Tech (CACHE)
WRIGHT AT MCMASTER	Joe Wright, McMaster University (CACHE)
WRIGHT.XRCC-NS AT XEROX.COM	Joe Wright, Xerox Research Centre, Canada (CACHE)
RAWSON at MILVM1	Norman Rawson, IBM Bethesda (CACHE)
MCUTLIP AT UCONNVM	Michael Cutlip, University of Connecticut (CACHE)
THOMAS AT UCONNVM	Thomas F. Anderson, University of Connecticut
SEADER at CC.UTAH.EDU	J. D. Seader, University of Utah (CACHE)
GVR at PURCHE	G. V. (Rex) Reklaitis, Purdue (CACHE)
RMAIL001 at NU	Richard Mah, Northwestern University (CACHE)
RUEYDER AT LSUCHE	Rueyder Jeng, LSU
CMREIB AT LSUCHE	Danny Reible, LSU
CORRIPIO AT LSUCHE	Armando Corripio, LSU
MCLAUGHL AT LSUCHE	Ed McLaughlin, LSU
GSULLIVA AT WATDCS	Gerald Sullivan, University of Waterloo
RCG360 AT MAINE	John Hassler, University of Maine
ROMIRA01 AT ULKYVM	Raul Miranda, University of Louisville
NUTTALL AT UNMB	H. E. Nuttall, Jr., University of New Mexico
HPREISIG AT TAMKBS	Heinz Preisig, Texas A&M
CYKJYCS AT NERVM	J. P. O'Connell, University of Florida
HALE AT CLEMSON	Jim Hale, University of Clemson
BRYAN AT FREMP11	Professor Bryan, Ecole des Mines, Paris, France
CROWE AT MCMASTER	Cameron Crowe, McMaster University, Canada
CERRLRL AT TECHNION	Ram Lavie, Technion, Israel
CERRANAN AT TECHNION	Avi Nir, Technion, Israel
CERZRZR AT TECHNION	William Resnick, Technion, Israel
PTC AT VIRGINIA [CSNET]	Peter Cummings, University of Virginia
EDELSON AT FSU	David Edelson, Florida State University
N1.JME AT ISUMVS	John Eggebrecht, Iowa State University

CACHE Bulletin Board Service

by Peter R. Rony
Virginia Tech

Objectives

As part of a contract with IBM, preliminary planning was completed for an electronic mail, bulletin board and distribution service in conjunction with the CACHE organization.

The project objective is to act as an active resource for the CACHE Electronic Mail Task Force by establishing and maintaining a computer conferencing and bulletin board service. The purpose of this report is to describe a plan for implementation of this service.

Background and Discussion

During the period November 1, 1986 through May 31, 1987, the GRAND teleconferencing software was selected for the CACHE bulletin board service, and contact was established in early March 1987 with the City University of New York (CUNY).

CUNY is authorized to sublicense GRAND and associated software under an agreement with IBM. After executing an appropriate license agreement, the GRAND software and the GRAIL and Cambridge Conferencing Interfaces were ordered on April 10, 1987. To encourage delivery of the software, several additional phone contacts with personnel at CUNY were made. The software arrived during the first week of June 1987.

Starting in June, the software will be installed and tested. 100 Megabytes of disk storage have been allocated as the initial configuration for the GRAND software, which requires approximately 30 megabytes, and CACHE bulletin board.

Upon testing and development of a better understanding of GRAND capabilities and potential, a mailing describing the CACHE bulletin board system and the GRAND software will be prepared and distributed to all member CACHE chemical engineering departments. The mailing should be complete by sometime in October 1987. The mailing will include, in addition to a description of the system and its purpose, a description of interfacing software required, an indication of how to access the system and get started, and the rules for use and software submission to the system. The rules will be patterned after the sample rules prepared by IBM.

A description of the system and rules for its use will also be sent (perhaps electronically) to all chemical engineering faculty who have submitted BITNET userids at the time of the mailing. Assisting in this endeavor will be the publication ***Chemical Engineering Faculties***, which plans to incorporate BITNET addresses (in addition to the normal phone and mailing addresses) in a future edition.

Specific plans for implementation and use of the CACHE bulletin board system will be detailed in a subsequent report. The bulletin board will presumably include both a software library and teleconferencing/electronic mail facilities. Use of both facilities will be initiated by LSU and interested CACHE faculty.

The key to success will most certainly be the collection of useful and desirable information in the bulletin board holdings. Public domain LSU software will be included in the software library immediately upon the installation of GRAND. This will be primarily micro and mainframe educational software including, for example, modeling and numerical analysis software that was developed for a junior level LSU course in the modeling of chemical engineering systems (ChE 2176). This software, and its implementation into the course, was described at a recent regional meeting of the American Society for Engineering Education.

Additional software that will be incorporated into the bulletin board library will be encouraged from other chemical engineering faculty. Description of recent CACHE projects are expected to be included. This will greatly improve their accessibility to other chemical engineering faculty and will significantly enhance the attractiveness of the bulletin board service.

As indicated above, the key to success will be the development of interesting and useful offerings on the bulletin board. Every effort

will be made to attract submissions of that quality once the GRAND system is installed and operational.

Implementation Plan

June 1987	Arrival and installation of GRAND teleconferencing software, which will act as the resident software for the bulletin board.
September 1987	Completion of GRAND testing. Development of a plan for the structure and usage of GRAND. Identification of GRAND user interface software requirements. Inclusion of available LSU software into the CACHE bulletin board, for example, ACS laboratories (from ChE 4198) and numerical analysis laboratories (from ChE 2176).
October 1987	Completion of mailing to all CACHE Chemical Engineering Departments describing the system and how to use it. A request for software submissions and the rules governing such submissions will be included.
January 1988	Inclusion into the bulletin board of all recent CACHE projects that exist in electronic form. Coordination with all current CACHE projects to ensure inclusion in the bulletin board and delivery in electronic form.

Group Purchase of Trillian Demo Generator Concurrent Authoring System

Peter R. Rony

During our 1987 Summer School for Chemical Engineering Faculty session, "Microcomputers in Laboratories," my colleague Ken Konrad and I demonstrated several demo diskettes—kindly provided by hardware vendors Data Translation and Metrabyte—that described laboratory data acquisition software such as Asyst, Labtech Notebook and DaDISP.

The Asyst demo diskette was particularly attractive, and prior to the meeting, I pursued the originator of the demo software—Trillian Computer Corporation in Los Gatos, California—to determine its price and availability to chemical engineering educators. The list price was staggering: $6,800 in quantities of one to an organization. Twenty-five copies could be obtained at a cost of $10,000 to a single organization. This latter price provided the software at a cost of $400 per copy, which is within the budgets of academic departments and research groups.

I asked Bill T. Sautter, Director of Product Marketing at Trillian, if an organization such as CACHE could purchase twenty-five copies for $10,000 and distribute them to twenty-five different member chemical engineering departments, who would pay $400 for each copy. While not committing to the $10,000 price, Mr. Sautter indicated that such a deal probably could be arranged if we could identify the twenty-five departments. Further, he provided twenty Trillian demo diskettes for distribution at the Summer School; these diskettes demonstrate the attractive features of the demo software.

Twelve department representatives at the Summer School have indicated interest in and have received a demo copy of the Trillian Demo Generator software, which runs on an IBM PC equipped with a CGA or EGA color monitor but no hard disk. We have eight demo copies left. If you would like to seriously consider purchasing the software package for about $400 and are interested in studying the demo diskette, please send an empty 5 1/4" disk mailer to me.

Why would a chemical engineering faculty member be interested in demo generator software? Gerry Beyer, a colleague in my department who has been working with such software for about a year, views it as a new and novel information delivery system for educators. For example, instead of providing a colleague with a printed manuscript, a "paper-on-a-disk," to use Gerry's term, can be disseminated. This disk would contain a sequence of images—text and graphics—controlled by demo generator software and presented in a visually attractive manner. Tables, graphs and a sampling of experimental data could all be contained on a single floppy disk.

Gerry demonstrated his single-disk "show" on magnetic lines of force to me only two weeks before the ASEE Summer School. As one who uses 2" x 2" slides and overhead transparencies in talks and courses, I was immediately attracted to the idea.

The demo generator software that he used—Showpartner—was easy to learn; it took only several hours to master the process of capturing CGA color screen images and to produce a script for a disk "show." I purchased my own copy of Showpartner, viewed the demo diskettes for Asyst, Labtech Notebook and DaD-ISP with renewed interest and pursued the demo generator from Trillian Computer Corporation, software that seemed considerably better than either Showpartner ($79) or Showpartner F/X ($350).

As you can imagine, a concurrent demo generator is an attractive way to deliver assistance to students concerning the use of software, laboratory equipment, and so forth. If you are interested in the Trillian software, please let me know immediately.

The product fact sheet and product announcement sheet that accompanied the Trillian demo diskettes distributed at the ASEE Summer School contained the following information:

Trillian Demo Generator Concurrent Authoring System

Four disks: System, Tutorial, Library, and Demo
Demo Generator reference manual
Concurrent Text Editor
Audio Text (TM), an optional cassette player/recorder for synchronized audio (interfaces to an IBM PC through a parallel printer port)
Remote Control Unit, an optional accessory that interfaces to an IBM PC through a parallel printer port
Graphics Editor and Mouse

Text-Oriented Capabilities

Interactively draw colorful windows with text and character graphics
Produce applications with a Concurrent Text Editor while live subject applications run in the background
Display pop-up windows and messages non-destructively on top of live subject application screens
Create custom menus, Lotus 1-2-3 like ring menus, and user interfaces
Compress text files to conserve disk space
Capture all or portions of screens
Fill screen areas with any character
Highlight and modify colors on screen
Obtain colors and attributes for any character on the screen

Read, write, and append to disk files and text-based libraries
Set screen border color
Set default screen colors and attributes
Map screen colors to specific monochrome attributes
Extract portions of text-based variables and files
Animation and special text effects
Concatenate strings for dynamic message creation

Bit-mapped Graphics-oriented Capabilities

Display pictures
Create and modify bit-mapped graphics pictures
Produce animated graphics and special effects
Multiple-font and clip-art libraries
Pop-up windows with animated scroll-in, scroll-out text and pictures
Superimpose text over graphics
Overlay graphics on top of existing pictures
Capture entire or portions of bit-mapped graphics screens
Extract portions of graphics screens and files
Measure coordinates and dimensions of pictures
Read and write graphics pictures to disk

Control Features

Run external MS-DOS programs from inside demo and tutorial shows
Define multiple "hot-keys" for precise product simulation, course interruption, and branching control
Branch to any location (unlimited)
Call local and global subroutines
Access local and global variables
Overlay multiple demo/tutorial sessions and disk volumes
Use IF-ELSE, REPEAT-UNTIL, WHILE, and other high-level constructs to control flow, execution, and iterative operations

Utilities

DRAW — text-based drawing, screen capture, and screen editing
CAPTURE — capture text- and bit-mapped graphics screens to disk
RULER — Measure bit-mapped picture coordinates
MENU — Create custom menus and user interfaces
HELP — Tool for creating on-line help systems
SOUND — Adds sound and music to demos and tutorials
LIBRARY — Routines supporting text libraries

System Control

Encrypt files to prevent user viewing and tampering
Automatically remove system from memory
Response Analysis and Keystroke Input/Output
Keystroke-by-keystroke with optional timeout
Buffered with multiple termination characters

Interpret keystrokes concurrently
Send keystrokes to subject application
Multiple input fields on a single screen
Record keystrokes for late playback and editing
Multiple correct answers and close answers
Interpret wildcard characters

Product Applications

Disk-based Demos with animation and interactivity for product
Concurrent and simulated Tutorials for product training and instruction
Prototypes and Screen Mock-ups for new software products
Computer-based Presentations for trade shows and product demonstrations
Concurrent and context-sensitive On-line Help and Documentation Systems to assist in product use, support, and training
Generation of product Installation Procedures and Scripts
Design of Custom Menus and User Interfaces for software (with optional mouse, touch-screen, and light-pen interfaces)
Software Testing and Quality Assurance
As a utility for producing High-level Macros and 'Super' Batch Files

* Lotus 1-2-3 is a trademark of Lotus Development Corporation. Showpartner is a trademark of Brightbill-Roberts and Company, Ltd. Asyst is a trademark of MacMillan Software Company. Labtech Notebook is a trademark of Laboratory Technologies Inc. DaDISP is a registered trademark. Trillian, AudioText, and Demo Generator are trademarks of Trillian Computer Corporation (Advanced Products Group, 405 Alberto Way, Mail Stop 1, Los Gatos, California 95030), (408) 358-2761.

GRAND SOFTWARE:
Visit to City University of New York Computer Center

by Michael D. Cutlip
Department of Chemical Engineering
University of Connecticut

On June 24, 1987, I visited the computer center at City University of New York (CUNY, 555 West 57th Street, 16th floor) to obtain more information on and a demonstration of the IBM GRAND system. The main contacts there are:

Mrs. Pat Reber, (203) 903-3622, the administrator in charge of the GRAND project.

Mr. George Lykos, (212) 903-3657 and GXLCU at CUNYVM, a software person who is the current GRAND administrator.

Mr. Ben Klein, a software person associated with the GRAND project.

I found that CUNY is working with IBM to make GRAND available to a limited number of universities including Delaware, LSU, Brown, Illinois at Chicago, Maryland, North Carolina at Chapel Hill, and Queens College.

My impression is that IBM is testing the waters to see if GRAND should be made into a product. The effort is relatively low level at CUNY, with no active programming. My impression is that several people at IBM are consulted by CUNY when needed for GRAND (Norm Rawson, CACHE trustee from IBM, indicates that GRAND is in wide use internally within IBM).

My discussion at CUNY established that CACHE is the first professional group that was considering using GRAND through BITNET, particularly for bulletin boards, conferencing and possible software distribution. It seems that CACHE is breaking new ground in this effort.

I was told that access to GRAND requires a user interface program on the user computer, CONFIR, for example, for full screen displays or GRAIL for TTY line editing. The user interface software costs about $300 per installation.

A GRAND conference administrator (for example, someone from CACHE to manage the bulletin board activity) needs to be a local user to the GRAND software. This could be accomplished through remote telephone dialup of CUNY or LSU or via a network like TIMNET.

CUNY seems to be very interested in CACHE activities. They are uncertain of the IBM relationships to LSU and CACHE. CUNY has a contract with IBM that only allows certain activities. The ability to access GRAND from any BITNET location is a major question for successful use by CACHE. I think that it is essential that GRAND users not be required to purchase additional software for their local machine. The CACHE GRAND conference administrator perhaps can communicate directly with the computer that has the GRAND software installed, but this is inconvenient.

GRAND has great potential, but major questions remain.

Progress Toward a Wide-Area Network For Chemical Engineers

Peter R. Rony

Department of Chemical Engineering
Virginia Tech

On March 3, 1987, Dr. Norman Rawson (IBM/Bethesda and CACHE), Mr. Larry England (IBM/Boca Raton), Dr. Edward McLaughlin (ChE Department, LSU), and Dr. Peter Rony (CACHE and ChE Department, Virginia Tech) met at Louisiana State University to discuss the implementation and testing of an experimental IBM software package, called GRAND, as a wide-area network file server to be located on a host mainframe in the chemical engineering department at Louisiana State University.

Information on the GRAND software is given in the references, copies of which will be sent to you if you write to the author of this report. To quote reference [1]: "GRAND is a system of cooperating virtual server machines and supplied applications running on various nodes throughout a network . . . GRAND supplies network-based applications such as computer conferencing and software distribution. In addition, GRAND provides a basis for running new applications. You can develop and add these to GRAND to either meet the particular needs of your installation, or the general needs of an entire network community. GRAND's services are available to all users throughout a network. You can use a GRAND server on your node (if there is one), or you can use a GRAND server on a neighboring node. In either case, you can conveniently use GRAND while logged on to your normal computer account."

Based on this description—which implies that chemical engineering faculty who have access to any BITNET node in the United States or abroad can use the GRAND at Louisiana State University—I have characterized this type of computer capability as a wide-area network file server.

Potential applications offered by GRAND that the chemical engineering community can explore include: computer conferencing, software distribution, mail and message sending, computer bulletin board, distribution of news items, centralized computer directories, centralized data repositories such as Who's Who lists, and so forth.

The GRAND software has been tested at one or two installations within IBM and at one New York institution (CUNY). It will not be sold by IBM as a product, but its features will be incorporated into future IBM products.

We are on the ground floor in the use of GRAND in academia.

Norm Rawson in particular, and others, including Dr. Vincent Vitagliano, at IBM should be thanked for providing this opportunity to the chemical engineering community. The existence of CACHE and its track record of service to academic chemical engineering, and the continuing interest of and support by CACHE trustees in electronic mail communications (despite our lack of success with COMPMAIL+), have been instrumental in their decision to go ahead with this experiment in wide-area computer networking within a profession.

GRAND Logistics: Questions and Answers

Details on the logistics of the installation, testing, and use of the GRAND software at LSU will now be presented in question-and-answer form.

Q: What action items were proposed by the participants at the March 3, 1987 meeting at LSU?

A: The installation of GRAND software in the chemical engineering department at LSU.

The testing of GRAND software, in conjunction with BITNET, in chemical engineering.

The hiring of a M.S. in computer science, Mr. Reyuder Jeng, to implement the installation of GRAND on the IBM mainframe that is being used for the ACS system in the chemical engineering depart-

ment at LSU. Mr. Jeng has already organized a bulletin board for the agricultural community at LSU.

The writing of a proposal on the installation, testing, and use of the GRAND software at LSU and the development of wide-area network file server capability for chemical engineering departments. The deadline for this proposal is June 1, 1987.

Q: What concerns were expressed about the use of GRAND software at LSU?

A: The possibility that it may prove to be too successful, and thus saturate the ACS mainframe in the chemical engineering department at LSU. The response to this concern was that initially, usage of GRAND within chemical engineering would probably be modest; the problem initially would be to encourage usage, not to curtail it. If the experiment proves to be successful, something that may occur only after several years of testing, it would be possible to distribute the computing load to other GRAND file servers on the network.

Also, if successful, (a) AIChE headquarters may become interested in wide-area networking for all chemical engineers, and (b) efforts by NSF to link all scientists and engineers within the United States electronically would be well underway, and would impact our efforts.

Q: What about "trash" on the system, in other words, improper use of GRAND.

A: The response to this concern was to have faith in the professionalism of faculty.

Q: What about costs? For what costs would CACHE be responsible?

A: Costs were not discussed at the LSU meeting on March 3, 1987. My impressions—and do not hold anybody to them—are that:

(1) IBM will provide the GRAND software to the LSU chemical engineering department at no charge; (2) IBM may be willing to finance the production of an attractive brochure, such as the ones for the ACS systems at Purdue and at LSU, on the use of GRAND once it has a modest track record within the chemical engineering community; (3) The costs that CACHE would presumably incur are in the category of information dissemination, namely, publicity, mailings to chemical engineering departments, and so forth.

All wide-area network communications will be handled over BITNET, the costs for which are distributed among the nodes on the BITNET network. If an individual department can access a local BITNET node, it is responsible for whatever costs that such access requires.

Q: What is the timetable for implementation and testing of GRAND?

A: Only three items in the timetable were discussed on March 3, 1987: (1) The hiring of Reyuder Jeng by the chemical engineering department at LSU; (2) The preparation of a preliminary planning report on the proposed GRAND wide-area network file server in chemical engineering by June 1, 1987; and (3) The presentation of this report to CACHE trustees at the Park City meeting in July 1987.

My conservative impressions for the likely timetable for this activity are as follows:

by June 1-15, 1987: Draft report in hands of CACHE Executive Committee

July 10-12, 1987: Discussion of report at CACHE meeting in Park City. Approval for this activity.

by Sep-Oct 1987: GRAND software installed at LSU; initial testing

by Spring 1988: First report on use of GRAND and BITNET in chemical engineering. Perhaps the availability of a videotape on the use of GRAND in ChE.

Q: Are there any opportunities for faculty members to become active in the CACHE Electronic Mail Task Force?

A: Yes, but wide-area networking will probably always be viewed by chemical engineering faculty as a necessary means to accomplish a useful end, and not as an end in itself. This is a service activity to the professional community. The Task Force welcomes one and all as partici-

pants in this exciting new communications activity.

Q: What problems with this activity can be anticipated?

A: The same problems that we, and others, have been experiencing in attempts to promote the use of wide-area networking: (1) Individual barriers to getting on the network (lack of hardware, lack of access to BITNET, lack of funds, lack of time or interest to master the software, and so forth); and (2) Lack of incentive to use the network (no reason to communicate with others on the network, no interest in the bulletin boards and computer conferences, and no interest in information available on the network).

The success of the GRAND wide-area network experiment will be based on our ability to identify services of sufficient attractiveness that our colleagues feel compelled to participate.

Potential examples include: (a) Centralized listings, and perhaps biosketches, of B.S. students who are interested in graduate school; (b) CACHE software and software tools that are available only in electronic form; (c) Computer conferences on timely subjects in chemical engineering; (d) Useful computer bulletin boards, and (e) the use of the IBM ACS within chemical engineering.

References

1 GRAND General Information Manual, Draft Copy—BITNET, IBM TJ Watson Research and Cambridge Scientific Centers, March 14, 1986.

2 R. A. Flavin, J. D. Williford, and H. Barzila, "Computer Conferencing Data Structures in the GRANDiose System," ***IEEE Trans. on Professional Comm.***, PC 29, 34 (March 1986).

Group Purchase of Texas Instruments' Personal Consultant Plus

Peter R. Rony

In their 1987 Summer School for Chemical Engineering Faculty session, "Expert Systems and Artificial Intelligence," Lyle Unger (University of Pennsylvania) and Mark Kramer (M.I.T.) presented the following brief overview of expert system shells:

Languages (alternative to shells)

OPS5, OPS83, Prolog, Lisp
Fortran, C, Pascal

Low End (Insight)

$100 to $500
Backward (and sometimes forward) chaining
Confidence factors

Middle Range (Personal Consultant Plus, Guru, Nexpert)

$500 to $10,000
Backward and forward chaining
More control over conflict resolution
Contexts
Better user interface
Translation of rules to "English"

High End (KEE, ART, KnowledgeCraft)

$30,000 to $70,000
More user interface
Graphical browsing
Much more than rules
Frames, inheritance, multiple worlds . . .
Takes longer to learn

Of the expert system shells and languages mentioned above, Personal Consultant Plus (PC Plus) was available to all participants during the session for a hands-on demonstration (on IBM PCs equipped with a hard disk) of how to use and modify a simple expert system program, for example, the selection of a printer for a personal computer. Thirty of us were sufficiently impressed with the ease of use of the software to wonder if a group purchase could be arranged at an advantageous price to educators. Both Lyle and Mark, as well as several others in the audience who already had obtained PC Plus, were optimistic that such a purchase could be arranged.

If you wish to add your name and institution to the list of thirty, please write to Peter Rony, Department of Chemical Engineering, Virginia Tech, Blacksburg, VA 24061. Do so immediately, within several weeks of receipt of this issue of CACHE NEWS. We will get back to you as soon as a decision is made.

You may have seen the two-page ads for Personal Consultant Plus in computer magazines. For example, on pages 16 and 17 of the September 1987 issue of ***Dr. Dobb's Journal,*** which just arrived, the following advertising information is provided:

Personal Consultant Plus 3.0 Standard Features ($2950)

Frames, rules, meta rules and procedures
Forward/backward chaining
Confidence factors
Regression testing and rule tracing
End-user explanation facilities
Graphics image capture and display
Interfaces to dBase, Lotus 1-2-3, DOS files, .EXE or .COM programs, "C"
Complete LISP development environment
2-Megabyte expanded/extended memory support
Mouse support
Context sensitive help
"Getting Started" tutorial-style manual

Personal Consultant Images ($495)

Optional add-on package to PC Plus (3.0)
Allows integration of "active images" into knowledge bases
Interactive dials, gauges, forms, and selection images
Multiple images can be combined on same screen
"Getting Started" tutorial-style manual

Personal Consultant Online ($995)

Optional add-on package for PC Plus (3.0)
ONLINE expert systems that interact directly with process data
Multiple interfaces to data acquisition and analysis programs
Knowledge base synchronization with process data
Functions for historical and predicted trends
Special user interface/reporting capabilities
"Getting Started" tutorial-style manual

As you can see, the software is expensive, generally out of the reach of many chemical engineering departments. This is the reason for our attempt at a group purchase.

* dBase is a trademark of Ashton-Tate. Lotus 1-2-3 is a trademark of Lotus Development Corp. Personal Consultant Plus is a trademark of Texas Instruments Incorporated.

CACHE IBM PC Lessons for Chemical Engineering Courses Other Than Design & Control

The Curriculum Task Force of the CACHE Corporation is completing the experimental phase of its project to prepare computer-based lessons for chemical engineering courses other than Design & Control. These lessons are intended to expose the students to "open-ended" problems and typical design alternatives. In October, departments that support CACHE received a complimentary set of the six lessons on separate diskettes. The table summarizes the lessons that were included.

Departments that do not support CACHE may purchase each lesson individually for $25 per lesson.

All of the lessons run on the IBM PC and may be installed on a shared network. Copies of the diskettes may be made for distribution to classes of students.

As indicated in our ***Instructions to Contributors***, a primary goal during this experimental phase is the introduction of open-ended problems throughout the core curriculum. The lessons are intended to give students an opportunity to deal with open-ended (i.e., "design") problems at all stages of the curriculum, and not just in the capstone Process Design course.

We would collectively or individually like to know your views and those of your colleagues with regard to the effectiveness and utility of these lessons. Some have been extensively tested, while other probably would benefit by additional testing of recently generated code. All have been under development for at least six months.

It is noteworthy that these lessons were contributed by the authors with little or no renumeration. CACHE is covering the cost of distribution.

We are considering the initiation of a more ambitious continuation project in which authors are paid to prepare lessons. We are also planning to distribute existing software. If you or faculty in your department would be interested in participating, please contact:

Prof. Warren D. Seider
University of Pennsylvania
Department of Chemical Engineering
Towne Building
220 South 33rd Street
Philadelphia, Pennsylvania 19104-6396

CACHE IBM PC Lessons

Lesson (Program)	Authors	Applicable for Courses in
Slurry Flow in Channels	B. Freeman, W. Provine, G. Dow & M. M. Denn, Berkeley.	Fluid Mechanics
Supercritical Fluid Extraction	J. Kellow, M. L. Cygnarowicz & W. D. Seider.	Separations and Thermodynamics
Gas Absorption with Chemical Reaction	K. Nordstrom & J. H. Seinfeld, Cal. Tech.	Separations
Design of Flash Vessels and Distillation Towers	B. A. Finlayson, E. W. Kaler & W. J. Heideger, Washington.	Separations and Thermodynamics
Heterogeneous Reaction Kinetics	J. E. Bauer & H. S. Fogler, Michigan.	Reactor Analysis
CSTR Dynamics and Stability	L. E. Vajdi & D. T. Allen, UCLA.	Reactor Analysis

ANNOUNCEMENT FROM ELSEVIER SCIENCE PUBLISHERS:

Computer Programming Examples for Chemical Engineers

by G. Ross

Dept. of Manufacturing Engineering
Swinburne Institute of Technology
Hawthorn, Vic., Australia

Written by a chemical engineer rather than by a computer scientist, this book fills the gap between texts which teach computer languages or programming methods and chemical engineering texts which omit details of writing programs. In order to write a computer program and get it to work, general theoretical principles are not enough; one has to actually do the job. This is done in each case by first taking the reader through a manual calculation, then presenting a computer program to perform the same task. Explanation of how the program operates is given in some detail.

Topics discussed in this way include: computer flowsheeting; interpretation and accessing of results and physical data; forward feed multi-effect evaporation; binary distillation; linear programming; introduction to finite differences with simple heat exchanger example; steady state multi-dimensional heat conduction; unsteady state heat conduction; solution of automatic control problems using finite differences. In each case, the necessary theory is fully introduced.

The programs are written in BASIC—an easy-to-learn, moderately powerful language available on both mainframe and desk-top computers. Now that microcomputers are so freely available, it is convenient for the reader to study the text and also follow the examples on the computer. To make this easier, the programs listed in the book are also available on diskette, ready to run on machines of the IBM PC range and compatibles. The programs presented are suitable for further development by the reader to suit his own requirements and the exercises at the end of each chapter are intended to encourage this. The book will thus be of interest to graduates, senior undergraduates and instructors of chemical engineering.

Contents: Introduction. 1. Flowsheeting (Process Simulation). 2. Interpretation and Accessing of Results and Physical Data. 3. Solution of Multiple Effect Evaporator Problems. 4. Solution of Distillation Problems. 5. Linear Programming. 6. Solution of a Counter-current Steady State Heat Exchanger. 7. Solution of Problems in Conduction Heat Transfer at Steady State, by the Method of Finite Differences. 8. Unsteady State Conduction of Heat. 9. Automatic Control. Subject Index.

ORDER FORM

Send to your bookseller or
ELSEVIER SCIENCE PUBLISHERS
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Distributor in the U.S.A. and Canada
ELSEVIER SCIENCE PUBLISHING CO., INC.
P.O. Box 1663, Grand Central Station, NY 10164-0028, U.S.A.

Please send me _____ copy(ies) of:

COMPUTER PROGRAMMING EXAMPLES FOR CHEMICAL ENGINEERS, by **G. Ross**
☐ Hardback: US$ 87.75 / Dfl. 180.00
☐ Paperback: US$ 48.75 / Dfl. 100.00
☐ Diskette: US$ 44.00 / Dfl. 90.00

Name _____

Address _____

_____ Postal Code _____

Orders from individuals must be accompanied by a remittance, following which books will be supplied postfree.

I enclose | my personal cheque | bank draft | UNESCO coupons |

Please bill my credit card: _____

American Express _____

Mastercharge/ Eurocard/Access _____

Issuing bank _____

Visa/Bankamericard/Barclaycard _____

Expiry Date _____

Signature Date _____

The Dutch guilder price is definitive. US$ prices are subject to exchange rate fluctuations. Prijzen zijn excl. B.T.W.

S & T/11P-147/15.000/0687/F

FLOWTRAN BOOKS

TO: ULRICH'S BOOKSTORE
Attention: E. Bundy
549 East University Avenue
Ann Arbor, MI 48109

Please send me:

FLOWTRAN SIMULATION-AN INTRODUCTION,
2nd Edition, by J.D. Seader, W.D. Seider, and A.C. Pauls
No. of copies _____ @ $13.95/each

EXERCISES IN PROCESS SIMULATION USING FLOWTRAN,
2nd Edition, Edited by J.P. Clark, T.P. Koehler, and J.T. Sommerfeld
No. of copies _____ @ $11.95/each

CACHE USE OF FLOWTRAN ON UCS,
2nd Edition, by R.R. Hughes and L.T. Biegler
No. of copies _____ @ $2.95/each

Add $1.50 per order for mailing and handling in U.S. A check must accompany all orders by individuals. Also available in quantity at regular quantity discounts to established book retailers.

Name: _____

Address: _____

_____

_____

_____
(City) (State) (Zip)

Make checks payable to Ulrich's Bookstore.

FLOWTRAN TAPE

☐ I am interested in **preparing** a FLOWTRAN tape.
☐ I am interested in **obtaining** a FLOWTRAN tape.

If you have checked either of the above squares, please complete the following information. If you have two computers you want to consider, duplicate this form, submit both completed forms and your preference.

1. Computer make and complete model number
   _____
2. Operating system version _____
3. FORTRAN compiler version _____
4. Magnetic tape facility:
   No. of tracks _____
   Drive speed in bits/inch _____

Name: _____

Address: _____

_____

_____

_____
(City) (State) (Zip)

Send the form to:

Professor J.D. Seader
CACHE
3062 MEB
University of Utah
Salt Lake City, UT 84112

Send this form only if you have not previously contacted Professor Seader.

PROCEEDINGS OF THE THIRD CHEMICAL PROCESS CONTROL CONFERENCE

I am interested in the Conference Proceedings published by Elsevier Science Publishers.

Please send ordering information to:

Name _____

Address: _____

_____

_____

_____

(City) (State) (Zip)

Mail this form to:
Elsevier Science Publishers
P.O. Box 211
1000 AE Amsterdam
The Netherlands

In U.S. & Canada:
Elsevier Science Publishing Co. Inc.
P.O. Box 1663
Grand Central Station
New York, NY 10163

CACHE PROCESS DESIGN CASE STUDY VOLUME 1

"Separation System For Recovery Of Ethylene And Light Products From A Naphtha Pyrolysis Gas Steam"

☐ $15 for initial copy of CACHE-supporting department _____

☐ $35 for additional copies _____

☐ $35 for non-supporting department _____

TOTAL _____

☐ Check for total amount is enclosed

☐ Purchase Order

Name _____

Address: _____

_____

_____

_____

(City) (State) (Zip)

Please Mail Order Form To:

CACHE Corporation
P.O. Box 7939
Austin, TX 78713-7939

Make check payable to CACHE Corporation.

CACHE Process Design Case Study Vol. 2
DESIGN OF AN AMMONIA SYNTHESIS PLANT

Preliminary Design and Economic Analysis

Summary: The objective of the case study is the design of an ammonia synthesis plant that is to be built in 1990, and that uses hydrogen and nitrogen feedstocks from a coal gasification plant. All stages of the design procedure starting from preliminary calculations down to the detailed flowsheet calculations are described. Emphasis is placed on the following steps: screening of key flowsheet decisions (pressure of synthesis loop, ammonia recovery, synthesis of gas recycle, hydrogen recovery from purge stream), selection of reactor configuration, cost minimization, and synthesis of heat exchanger network.

The proposed design incorporates a medium-pressure synthesis loop with water absorption/distillation for ammonia recovery, and with membrane separation for hydrogen recovery. The process was designed with the simulator PROCESS from Simulation Sciences, and the ammonia reactor was designed with the special purpose package QBED. A listing of this program is included in the case study. Depending on the required detail and the availability of process simulation software, the case study is suitable as a one-term assignment for a single student or a group of students. The preliminary calculations of the case study were performed by a group of three students, while the final design report is based on the work of a group of five students.

The problem statement was supplied by Philip A. Ruziska from Exxon Chemicals, and the case study was prepared under the supervision of Ignacio E. Grossmann from Carnegie-Mellon University.

CACHE PROCESS DESIGN CASE STUDY VOLUME 2

"Design of an Ammonia Synthesis Plant"

☐ $15 for initial copy of CACHE-supporting department _____

☐ $35 for additional copies _____

☐ $35 for non-supporting department _____

TOTAL _____

☐ Check for total amount is enclosed

☐ Purchase Order

Name _____

Address: _____

_____

_____

_____

(City) (State) (Zip)

Please Mail Order Form To:

CACHE Corporation
P.O. Box 7939
Austin, TX 78713-7939

Make check payable to CACHE Corporation.

CACHE Process Design Case Study Vol. 3
DESIGN OF AN ETHANOL DEHYDROGENATION PLANT

Preliminary Design and Economic Analysis

Summary: The objective of this case study is the preliminary design of an acetaldehyde synthesis process by ethanol dehydrogenation. The project covered all stages of the design procedure starting from consideration of qualitative aspects of the flowsheet and preliminary calculations to detailed process simulations and final economic evaluations. In this study emphasis is placed on synthesizing a workable flowsheet and justifying its configuration, simulating and evaluating the design using a commercial process simulator, and deriving a heat recovery network for the final process.

The main reaction in this process is the endothermic dehydrogenation of ethanol to acetaldehyde. However, under the specified reactor conditions, a number of byproducts are produced and their presence determines a number of interesting alternatives for separation. Once these alternatives have been screened and a workable flowsheet has been synthesized, the study centers on the simulation of this flowsheet using PROCESS from Simsci, Inc. Here, some of the features, advantages and limitations of this simulator are presented. Finally, the study concludes with a complementary presentation of this process simulated with the CACHE version of FLOWTRAN. While the aim of this study is not to provide a detailed comparison between PROCESS and FLOWTRAN, a useful description of the relative merits of both simulators can be readily observed.

This project is suitable for a one-term project by a five or six person team of senior design students. The results of two such teams are given in this study.

This problem was posed by the Union Carbide Corporation and the case study was prepared under the supervision of L.T. Biegler of Carnegie-Mellon University and R.R. Hughes of the University of Wisconsin.

CACHE PROCESS DESIGN CASE STUDY VOLUME 3

"Design of an Ethanol Dehydrogenation Plant"

☐ $15 for initial copy of CACHE-supporting department _____

☐ $35 for additional copies _____

☐ $35 for non-supporting department _____

TOTAL _____

☐ Check for total amount is enclosed

☐ Purchase Order

Name _____

Address: _____

_____

_____

_____

(City) (State) (Zip)

Please Mail Order Form To:

CACHE Corporation
P.O. Box 7939
Austin, TX 78713-7939

Make check payable to CACHE Corporation.

Order Form for MicroCACHE Software/Documentation

(enter amount to indicate order)

Basic Package $ 400. _____

Software:

1. MicroCACHE Module Presentation (**MSM**) Diskette
2. MicroCACHE Module Preparation/Information Retrieval (**MCU**) Diskette
3. MicroCACHE McCabe-Thiele **MOD**ule (MCCABE) Diskette
4. MicroCACHE McCabe-Thiele Module **Source** File Diskette

Documentation:

1. *Overview*
2. *User's Manual*
3. *Instructor's Manual*
4. *Module Writer's Manual*
5. *Graphics Packages Manual*

Videotapes (VHS Format):

1. *The MicroCACHE CAI System*
2. *Module Preparation I*
3. *Module Preparation II*

Modules

1.	a) Interactive Graphical Flowsheet Preparation b) Steady-State Process Simulator (Mat. Bal. Only) (CLASS)	$ 150.	_____
2.	Material Balance Exercises (MATBAL)	$ 90.	_____
3.	Binary Distillation - Ponchon-Savarit Method (PONCH)	$ 90.	_____
4.	Binary Batch Distillation (BATCH)	$ 90.	_____
5.	Solution of Single Nonlinear Equations* (NLEQN)	$ 90.	_____
6.	Numerical Integration* (ROMBRG)	$ 90.	_____
7.	Solution of Systems of Linear Equations (SLINEQ)	$ 90.	_____
8.	Solution of Systems of Nonlinear Design Eqns.* (NEWRAP)	$ 90.	_____
9.	Solution of Systems of Ordinary Diff. Equations* (RKFOUR)	$ 90.	_____
10.	Linear Programming (LINPRO)	$ 90.	_____
11.	Nonlinear Programming (Dav.-Fletch.-Powl./Broyd)* (UNLP2)	$ 90.	_____
12.	Gas Reservoir Simulation (GASSIM)	$ 90.	_____

TOTAL _____

*Module requires that a copy of the MicroSoft FORTRAN-77 compiler be available. These modules should be used only on PCs with hard disks (e.g., XT,AT).

Please **sign** the reproduction/distribution **agreement** on the reverse side, **enclose purchase order** or **check** (payable to CACHE) for total amount, and return to Prof. Carnahan.

Agreement for Reproduction/Distribution of MicroCACHE Software and Documentation

CACHE, Inc. (a nonprofit educational corporation) holds the copyright to the MicroCACHE system software, documentation, and educational modules distributed under CACHE auspices. MicroCACHE software is not copy-protected, hence can be easily reproduced and distributed to your faculty and students. Placing an order for purchase of software and documentation implies agreement to the following conditions with regard to copying and distributing MicroCACHE materials.

The purchased MicroCACHE software is to be used by students, faculty, or module authors within a **single department** only. Permission is given by CACHE to:

1. Reproduce as many copies of the **MCU** diskette **files** as there are departmental **instructors** or **authors** using or preparing MicroCACHE modules for departmental students.

2. Reproduce as many copies of the **MSM** diskette **files** as there are **students** using MicroCACHE modules.

3. Reproduce as many copies of **MOD** ule diskette **files** as there are **micro-computers** accessible to students; distribution from a local area network file server **is** allowed.

4. Reproduce **module source files** as needed by **instructors** for module modification.

5. Reproduce **documentation** as follows:

Overview	unlimited
User's Manual	for students/instructors/authors
Instructor's Manual	for instructor/authors
Module Writer's Manual	for instructors/authors
Graphics Packages Manual	for instructors/authors
Videotapes	for instructors/authors

6. With the exception of the *Overview* document, permission to reproduce software or documentation for use **outside** the purchasing department is **withheld** (i.e., each department should purchase its own copy of copyrighted MicroCACHE materials).

Please **sign** below to accept these terms and return with your **order** on the **reverse side** of this sheet to:

Prof. Brice Carnahan
Chemical Engineering Dept.
The University of Michigan
Ann Arbor, MI 48109

_____ _____
signed date

for: department _____
institution _____
address _____
_____

CAST/CACHE

Foundations of Computer-Aided PROCESS DESIGN

Editors: **Arthur Westerberg**, Carnegie-Mellon Univ.; **Henry H. Chien**, Monsanto Company

This 1050 page hard-bound volume features 22 papers by leading academic and industrial researchers in the computer-aided design field. The major topics (**see backside for complete list of papers and authors**) are:

1. Expert Systems
2. Overview and Outlook for Process Systems Engineering
3. Progress in Data-Base Development
4. Computational Algorithms
5. Physical Properties for Design
6. Nonsequential Modular Flowsheeting
7. Design and Sequential Scheduling of Batch Chemical Plants
8. Complex Single-Unit Design (columns, reactors)
9. Operability in Design
10. Contributed Papers on Scheduling, Property Prediction, and Nonlinear Equation Algorithms

---

Please send me ____ copies of Foundations of Computer-Aided Process Design @ $ 37.50 postpaid (make check payable to CACHE). Total $ _____

Name _____
Address _____
City _____ State ____ Zip _____

Send form and remittance (or purchase order) to:

CACHE CORPORATION
P.O. BOX 7939
AUSTIN, TEXAS 78713

Foundations of Computer-Aided Process Design

Contents

LIST OF CHEMICAL ENGINEERING DEPARTMENTS SUPPORTING CACHE

CACHE annually solicits universities for funds to carry out on-going CACHE activities and nurture new projects. The following is a list of our generous supporters:

1987-89

Florida A&M University
University of Akron
State University of New York, Buffalo

1986-88

University of Alabama, Tuscaloosa
Tuskegee Institute
University of California, Berkeley
California State Polytechnic University, Pomona
Stanford University
University of Colorado
Colorado School of Mines
Howard University
Georgia Institute of Technology
University of Idaho
Illinois Institute of Technology
Northwestern University
Purdue University
Rose-Hulman Institute of Technology
University of Iowa
Iowa State University of Science & Technology
University of Kansas
University of Kentucky
Louisiana State University
University of Southwestern Louisiana
University of Maine
Johns Hopkins University
Massachusetts Institute of Technology
University of Michigan
Michigan Tech. University
University of Minnesota
Mississippi State University
Dartmouth College
University of New Hampshire
New Jersey Institute of Technology
Princeton University
City College of City University of New York
Cornell University
North Carolina A&T State University
University of North Dakota
Ohio State University
Ohio University
Youngstown State University
University of Oklahoma
University of Tulsa
University of Pennsylvania
University of Pittsburgh
Widener University
Brown University
University of Houston
Virginia Polytechnic Institute & State University
Washington State University
West Virginia College of Graduate Studies
West Virginia Institute of Technology
University of Wyoming
Laval University
McMaster University
University of New Brunswick
University of Ottawa
Ecole Polytechnique—University of Montreal
Queen's University
Royal Military College of Canada
University of Saskatchewan
University of Waterloo
University of Toronto
University of Windsor
University of Sydney
Ecole Nationale Superieure

LIST OF INDUSTRIAL CONTRIBUTORS TO CACHE

The following companies have contributed financial support to specific CACHE activities during 1986-87:

Chevron Research Corporation
DuPont Committee on Educational Aid
Exxon Educational Foundation
Shell Companies Foundation
Xerox Foundation

ANNOUNCING

PSE '88

Third International Symposium on Process Systems Engineering

Sydney, Australia

August 28 through September 2, 1988

This Conference is being sponsored by the Institution of Chemical Engineers in Australia, and the Chemical Engineering College of the Institution of Engineers, Australia, on behalf of the Asian Pacific Federation of Chemical Engineering, and the Inter American Federation of Chemical Engineering. It is the third in a triennial series entitled PSE, and follows highly successful events held in Kyoto, Japan in 1982, and in Cambridge, England in 1985.

The Conference is affiliated with CHEMECA 88, a major international version of the annual meeting of Australasian chemical engineers. CHEMECA 88, an approved event of Australia's Bicentennial Year, and PSE '88 will share a common opening session, and two other plenary sessions, where topics of importance to the chemical engineering community will be addressed by speakers of international standing.

Six technical sessions are planned, each conducted by a Chairman-Rapporteur, containing presentations of five to six papers of 30 minutes duration, including discussion. Following the successful poster session at PSE '85, a similar session is planned this time. The Conference proceedings will be published. An exhibition relevant to the themes of the Conference will run concurrently.

Main Conference Themes

Process Control & Optimization
Artificial Intelligence
Batch Process Design & Operation
Industrial Applications
Failure Analysis in Design
Design of Flowsheets
Modelling
Education in PSE

If you are interested in attending or would like more information, please send your name and address to the address below. Please let us know also if you are interested in submitting a paper, and attach an abstract.

PSE '88 Conference
The Institution of Engineers, Australia
11 National Circuit
Barton, ACT 2600
Australia