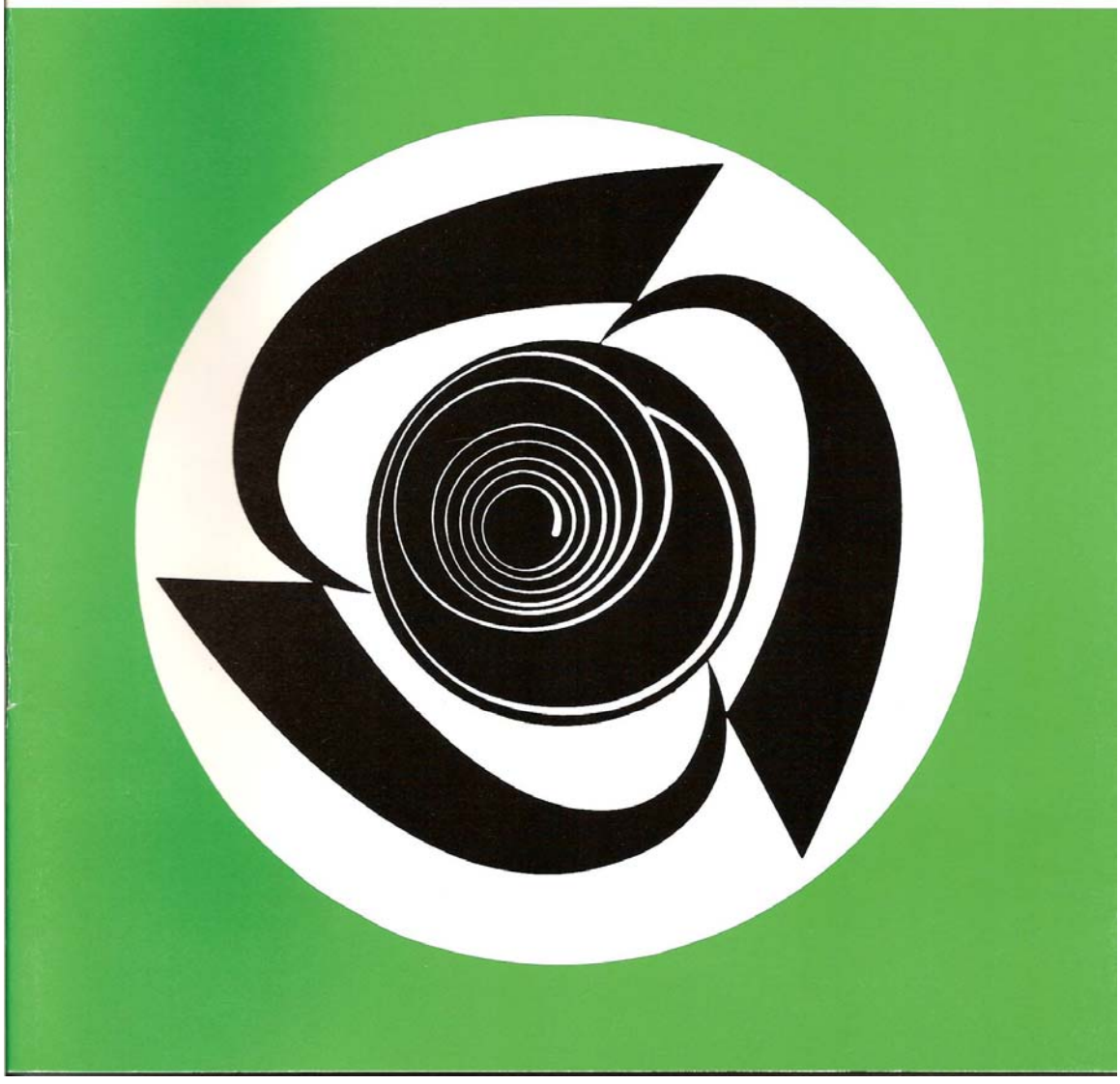
CACHE NEWS

NEWS ABOUT COMPUTERS IN CHEMICAL ENGINEERING EDUCATION

No. 40

Spring 1995



THE CACHE CORPORATION

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 (ComputerAids for Chemical Engineering) Committee. The CACHE Committee was initially sponsored by the Commission on Education of the NationalAcademy 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
Fax (512) 295-4498
CACHE@UTS.CC.UTEXAS.EDU.

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 Margaret Beam with contributions from a number of CACHE members and representatives.

CACHE NEWS

No. 40 Spring 1995

CACHE Trustees

President:
Warren D. Seider, University of Pennsylvania
(SEIDER@CHEME.SEAS.UPENN.EDU)

Vice President
Lorenz T. Biegler, Carnegie Mellon University
(BIEGLER@CMU.EDU)

Secretary
James F. Davis, Ohio State University
(DAVIS@KCGL1.ENG.OHIO-STATE.EDU)

Executive Officer
David M. Himmelblau, University of Texas at Austin
(CACHE@UTS.CC.UTEXAS.EDU)

Academic Trustees
Yaman Arkun, Georgia Institute of Technology
(YAMAN.ARKUN@CHEMENG.GATECH.EDU)
Brice Carnahan, University of Michigan,
(BRICE.CARNAHAN@UM.CC.UMICH.EDU)
Michael B. Cutlip, University of Connecticut
(MCUTLIP@UCONNVM)
Michael F. Doherty, University of Massachusetts
(MDOHERTY@ECS.UMASS.EDU)
Thomas F. Edgar, University of Texas at Austin
(EDGAR@CHE.UTEXAS.EDU)
H. Scott Fogler, University of Michigan
(H.SCOTT.FOGLER@UM.CC.UMICH.EDU)
Ignacio Grossmann, Carnegie Mellon University
(GROSSMANN@CMU.EDU)
Andrew N. Hrymak, McMaster University
(HRYMAK@MCMAIL.CIS.MCMASTER.CA)
Jeffrey Kantor, University of Notre Dame
(JEFFREY.C.KANTOR.1@ND.EDU)
Sangtae Kim, University of Wisconsin
(KIM@CHEWI.CHE.WISC.EDU)

Academic Trustees, continued

Richard S. H. Mah, Northwestern University
(MAH@CCADMIN.TECH.NWU.EDU)
Manfred Morari, ETH Zurich
(MORARI@AUT.EE.ETHZ.CH)
Gintaras V. Reklaitis, Purdue University
(REKLAITI@ECN.PURDUE.EDU)
Peter R. Rony, Virginia Polytechnic Institute and State University
(RONY@VTVM1.CC.VT.EDU)
J. D. Seader, University of Utah
(SEADER@UTAHCCA)
George Stephanopoulos, Massachusetts Institute of Technology
V. Venkatasubramanian, Purdue University
(VENKAT@ECN.PURDUE.EDU)

Industrial Trustees
Joseph Boston, Aspen Tech
(BOSTON@ASPENTEC.COM)
Carlos E. Garcia, Shell Development Co.
John C. Hale, E. I. DuPont de Nemours & Co.
(HALE@PCCVAX.DNET.DUPONT.COM)
Ferhan Kayihan, Weyerhaueser
(FKAYIHAN@WDNI.COM)
Edward M. Rosen, EMR Technology Group
(C00820ER@WUVMD)
Jeffrey J. Siirola, Eastman Chemical Co.
(SIIROLA@EMN.COM)
Joseph D. Wright, Pulp and Paper Research Institute of Canada
(WRIGHT@PAPRICAN.CA)

Advisory Committee
Lawrence B. Evans, Aspen Technology
James R. Fair, University of Texas at Austin
Thomas H. Lafferre, Monsanto Company
John J. McKetta, University of Texas at Austin

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 at Austin
Austin, Texas 78712
(512) 471-7445

Development
Dr. Thomas F. Edgar
Dept. of Chemical Engineering
University of Texas at Austin
Austin, Texas 78712
(512) 471-7995

Conferences
Prof. Richard S. H. Mah
Dept. of Chemical Engineering
Northwestern University
Evanston, Illinois 60201
(708) 491-5357

TASK FORCES

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

Process Engineering
Dr. Edward M. Rosen
Monsanto Company - F2WK
800 N. Lindbergh Blvd.
St. Louis, Missouri 63167
(314) 694-6412

Case Studies
Prof. Manfred Morari
Chemical Engineering 210-41
California Institute of Technology
Pasadena, California 91125
(818) 395-4186

Curriculum
Prof. James F. Davis
Dept. of Chemical Engineering
Ohio State University
140 West 19th Avenue
Columbus, OH 43210-1180
(614) 292-2359

Electronic Mail/CD Development
Prof. Peter Rony
Dept. of Chemical Engineering
Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061
(703) 231-7658

NSF Development of Innovative Engineers
Prof. H. Scott Fogler
Dept. of Chemical Engineering
3168 Dow Building
University of Michigan
Ann Arbor, Michigan 48109
(313) 764-2384

NSF Simulated Laboratory Modules
Prof. Gintaras V. Reklaitis
School of Chemical Engineering
Purdue University
West Lafayette, Indiana 47907
(317) 494-4075

Thoughts About CACHE - 25 Years Later

By Warren D. Seider, University of Pennsylvania, CACHE President

In July, 1994, at the CACHE Meeting in Snowmass, I was re-elected CACHE President, the first of the thirteen presidents to serve a second, non-contiguous term. Due to prior obligations, I could not assume the responsibilities of President until January 1. Indeed, we were fortunate in that Mike Cutlip agreed to extend his two-year term for another six months through December. The CACHE Trustees, and the chemical engineering faculty at large, owe Mike a great vote of thanks for his leadership in many respects. Mike's dedication has resulted in a host of achievements over the past two and one-half years.

As my second term begins, I cannot help but reflect on the remarkable development of CACHE in just 25 years. This is due to the synergistic effects of many educators and representatives from Bindustry working together to advance the usage of computers in chemical engineering education. In 1969, the challenges were to establish viable projects and to find a home for CACHE with financial support so as to enable the chemical engineering faculty, at many different institutions, to cooperate. After establishing a home in the Commission on Education of the National Academy of Engineering and obtaining funds from the NSF, our efforts were concentrated on the creation of an organizational structure, with Bylaws, the development of task forces, and the evolution of a distribution mechanism for newsletters, reports, and computer programs. Twenty-five years later, after the distribution of over 70 publications, most of which were created by the members of CACHE, in cooperation with other educators and people from industry, CACHE is a mature organization. These publications include reports, the results of surveys, computer programs, and conference proceedings, to mention a few.

It is most noteworthy, in my opinion, that CACHE has achieved so much success while operating on a relatively small budget (currently on the order of $80K/yr.) Under the able direction of David Himmelblau, the CACHE Office operates with part-time assistants. At no time in the history of CACHE have full-time staff persons been employed. The overriding majority of contributions have been accomplished by the CACHE trustees and their colleagues. Some of the work has been funded by the NSF, but much has been performed voluntarily. The contributions have come from outstanding faculty, who concentrate on teaching and research and are usually not distracted by large administrative responsibilities, such as arise in connection with for-profit companies.

As computing has become more pervasive in the chemical engineering curriculum, the role of CACHE and its impact has grown significantly. Each year CACHE has hosted a reception at the annual meeting of the AIChE. In San Fransisco, in connection with our 25th Anniversary Celebration, our reception was the best ever! Hundreds of chemical engineers, from academia and industry, filled a ballroom. They were able to try out many of the latest CACHE products, to view the CACHE Time-line (with over 70 publications on display), and to receive a free copy of the CACHE 25th Anniversary CD-ROM. The latter, prepared by Peter Rony with a generous donation by the 3M Prerecorded Optical Media in Menomenie, Wisconsin (who prepared 1200 copies at no cost to CACHE), contains 680 megabytes involving selected CACHE software, descriptions of nine chemical engineering departments, chemical engineering software demos (e.g., ASPEN PLUS, HEXTRAN, HTRI software,...), Internet Client Software, and much more.

Plans for 1995

CACHE has a very full plate of activities for 1995. If you are not yet involved, your participation is welcome. There are many ways to contribute, ranging from the preparation of software and teaching materials, to the distribution of previously prepared software, to the participation on a task force or as a CACHE Associate, to the preparation of an article for the CACHE NEWS, and many others. Below is a list of our plans. As you skim through them, perhaps you will see an opportunity to make a contribution. If so, please contact me directly or the person immediately involved.

1. Complete the 25th Anniversary Monograph. Brice Carnahan expects to have it printed and distributed in February. It will be distributed free to all chemical engineering faculty.

2. Evaluate the 25th Anniversary CD-ROM, with the possibility of creating a CD-ROM Task Force which

might issue, annually, an inexpensive CD-ROM for students and faculty. Peter Rony needs help and, better yet, a person to take over the preparation of the next CD-ROM, as he concentrates on the next activity.

3. Begin an electronic communications project. Emphasis here will be on the creation of a World Wide Web server.

4. Distribute the Chemical Reactor Design Toolbox, prepared by the NSF/CACHE Project at the University of Washington, under the direction of Bruce Finlayson.

5. Publish the Proceedings of the FOCAPD'94 Conference early in 1995. These will be distributed by the AIChE.

6. In cooperation with the CAST Division of the AIChE, facilitate the first ISPE Conference in Snowmass (July, 1995).

7. In cooperation with the CAST Division of the AIChE, plan the CPC V Conference, to be held in January, 1996.

8. Update the position paper, "Expectations of the Competence of Chemical Engineering Graduates in the Use of Computing Technology." This will be accomplished by the Curriculum Task Force and submitted to the Education and Accreditation Committee of the AIChE.

9. Organize the CACHE Industrial Associates. In this way, we hope to stimulate more usage of CACHE products in industry and more industrial involvement in all phases of the use of computers in chemical engineering education.

10. Launch a task force on statistics, with Richard S. H. Mah as chair.

11. Complete another Process Design Case Study involving the main fractionator associated with a hydrocracker. This case study is being prepared by Andy Hrymak.

12. Begin the preparation of a Process Design Case Study with emphasis on pollution prevention or minimization.

13. Continue the development of PICLES, POLYMATH, and CHEMSEP, three of the CACHE software packages.

14. Continue to distribute new modules prepared by the NSF/CACHE projects on the Development of Innovative Engineers (at the University of Michigan) and on the Simulated Laboratory Modules (at Purdue University.)

Again, there are many opportunities to participate. Please be in touch with us on these or projects of your own choosing.

INDUSTRIAL CONTRIBUTORS TO CACHE

The following companies have recently contributed financial support to specific CACHE activities:

Dow Elanco
DuPont Committee on Educational Aid
Tektronix
3M
Xerox Foundation

Picles 4.1 and the Case of the Interacting Controllers

By Douglas Cooper, University of Connecticut

Introduction

Picles™, the Process Identification and Control Loop Explorer System, is IBM PC compatible software now being used in more than 65 process dynamics and control courses around the world. Picles is an easy-to-use training simulator that provides hands-on experience to those studying this often abstract and mathematical subject. With over a half-dozen companies now using the software for employee training, Picles truly offers "real-world" experiences to students.

Picles contains a series of case studies, animated in color-graphic display, for self-paced or faculty guided learning. Users can manipulate process variables in open loop to obtain pulse, step, sinusoidal or ramped test data. Picles can record this data as printer plots or disk files for process identification and controller design. Digest™, companion software to Picles, is one package well suited for this identification and design task. After designing a controller, return to Picles and immediately evaluate and improve upon the design for both set point tracking and disturbance rejection.

A beta test version of Picles version 4 is being used in about 25 classrooms this spring. The new version retains all capabilities of the previous release and adds a host of new ones. Major additions include a multiple steady state jacketed CSTR process which can be operated either in a single loop control mode or a cascade control mode; a digital controller for studying dead-beat, Dahlin and other z-domain algorithms; an improved data plotting facility to make homework solutions more convenient to document; and an expanded on-line help facility.

After a brief review of Picles' features, this article explores one case study possible with the distillation column process. The distillation column has both a distillate and bottoms concentration control loop which, when implemented in the traditional manner, will interact and thus seriously degrade overall performance. The design of model based decouplers is then presented which work to mitigate this interaction. As part of the discussion, the use of the Digest dynamic modeling and controller design package is also demonstrated.

The Picles Case Studies

Previous CACHE Newsletters describe the Picles processes in some detail. Picles version 4 adds a multiple steady state jacketed CSTR process with a single loop or cascade control configuration to yield the case study options:

One-Input One-Output Case Studies:
Gravity Drained Tanks, Heat Exchanger, Pumped Tank, Mystery Processes

Ideal Transfer Function Case Study:
Design a Process

Multiple Steady State Case Study:
Jacketed Reactor

Two-Input One-Output Cascade Case Study:
Jacketed Reactor

Two-Input Two-Output Multivariable Case Study:
Distillation Column

The Picles Controllers

The Picles control algorithms can be custom tuned and implemented in only a few key strokes. Available controllers, which permits a broad variety of concepts to be explored, include:

Manual Control
P-Only Control
I-Only Control
Velocity PID Control with Derivative on Measurement
Velocity PID Control with Derivative on Error
Position PID Control (no windup protection)
Velocity PID with Smith Predictor
Velocity PID with Feed Forward
Velocity PID with Decouplers
Digital Sampled Data Controller

The Distillation Column Process

The Distillation Column Process, shown in Fig. 1, is a binary distillation column that separates water and methanol. The column dynamics are simulated using a model described by Wood and Berry (1973).

As shown, there are two measured/controlled variables and two manipulated variables for this process. The reflux rate controls the top distillate concentration and the rate of steam to the reboiler controls the bottoms concentration. The feed rate to the column is the disturbance variable.

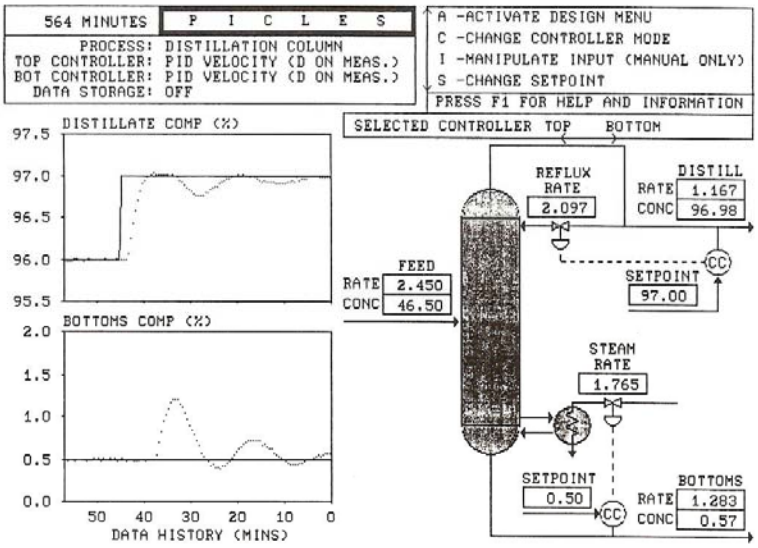


Figure 1 - Interaction of the Top and Bottom Control Loops in Distillation Column

This process illustrates the interaction which can occur in some multi-controller applications. For example, suppose the concentration (or purity) of product out of the top of the column is too low. The top controller will compensate by increasing the flow of cold reflux into the column. This increased reflux flow will indeed increase the concentration of the product out of the top of the column. However, the additional cold reflux will work its way down and eventually begin to cool the bottom of the column. This cooling will cause the concentration out of the bottom of the column to move off set point and produce a controller error.

The bottom controller will compensate by increasing the flow of steam into the reboiler. This produces an increase in hot vapors traveling up the column, which eventually causes the top of the column to begin to heat up. The result is that the concentration of product out of the top of the column again becomes too low. In response, the top controller compensates by further increasing the flow of cold reflux into the column. The controller "fight" begins. Decouplers are simple models of the process which can be designed into a controller architecture to minimize this interaction.

Objective of This Case Study

In this case study, the top and bottom controllers are first designed and implemented individually. The objective chosen here is a PI controller which can efficiently track set point steps between a top (distillate) concentration of 96% and 97%. The bottoms PI controller is to track set point steps between a bottoms concentration of 0.5% and 1.0%.

When both PI controllers are on-line simultaneously, they are shown to interact and degrade overall performance. The design of decoupler models incorporated as part of the controller architecture is then presented. Finally, the implementation of the decouplers is shown to succeed in dramatically reducing this undesirable interaction. Note that a host of other studies could also be explored using the options available in Picles.

Top (Distillate) Concentration Control

Given the above objective, the traditional method of approach to feed back controller design as detailed in the popular process control texts is to:

- step or pulse the manipulated variable (reflux rate) in open loop (manual mode),
- record the manipulated and measured variable data as the process responds to the step or pulse,
- fit a low order linear dynamic model to this manipulated to measured variable data,
- use the linear dynamic model parameters in one of several popular correlations to obtain initial estimates of controller tuning parameters,
- implement this controller (close the loop) and evaluate its performance in tracking set points (servo response), and rejecting disturbances (regulatory response),

- perform a final tuning by trial and error until desired controller performance is obtained.

Following this procedure for designing the top concentration controller, a pulse test is performed by stepping the reflux rate from its original steady state value of 1.95 lb/min up to 2.03 lb/min with both top and bottom controllers in manual mode. These values of the manipulated reflux rate are chosen as they cause the measured top concentration to change in manual mode from a concentration of 96% up to the stated objective concentration of 97%. After approximately 50 simulation minutes, the reflux rate is stepped back down to the original value of 1.95 lb/min. Using Picles' file save facility, the process data is saved to disk using the default sample rate.

There are a host of methods for fitting a low order dynamic model to the data collected from this dynamic experiment. These range from simple graphical methods such as Cohen-Coon to sophisticated analysis and regression packages such as MATLAB. In this case study, Digest, a companion product of Picles, is used to quickly and conveniently perform this model fitting task.

Using DIGEST for Dynamic Modeling

Digest is capable of importing ASCII files containing dynamic data from Picles, other software packages and even from a real plant. The data must be in ASCII tabular form with data columns separated by tabs, commas or spaces. Simple commands are used to mark which data column contains the manipulated process data, which contains the measured process data, and which contains the time data. The linear models available in Digest include first order, first order plus dead time (FOPDT), second order and second order plus dead time dynamic forms.

Digest then fits the process gain, time constant(s) and dead time (if applicable) to the data by minimizing the sum of the squared error (SSE) between the actual measured response and the predicted model response when using the manipulated variable process data contained in the file. In computing the SSE, Digest operates according to the assumptions:

- the process is at steady state before the dynamic event occurs,
- the first data point in the file is a good median value of the initial steady state, and
- the time increment between the data points is constant.

Fig. 2 shows the Digest fit of a FOPDT model to the dynamic data collected from the reflux rate pulse test just described. In this case, a FOPDT dynamic model approximates the process data extremely well as the model and measured data are virtually indistinguishable. As shown, the FOPDT model parameters computed by Digest are (the subscript TR stands for Top Concentration - Reflux Rate):

$$\text{Process Gain, } K_{P,TR} = 12.79 \ \frac{\%}{\text{lb / min}}$$

Overall Time Constant, $\tau_{P,TR} = 16.29$ minutes

Apparent Dead Time, $\theta_{TR} = 1.336$ minutes

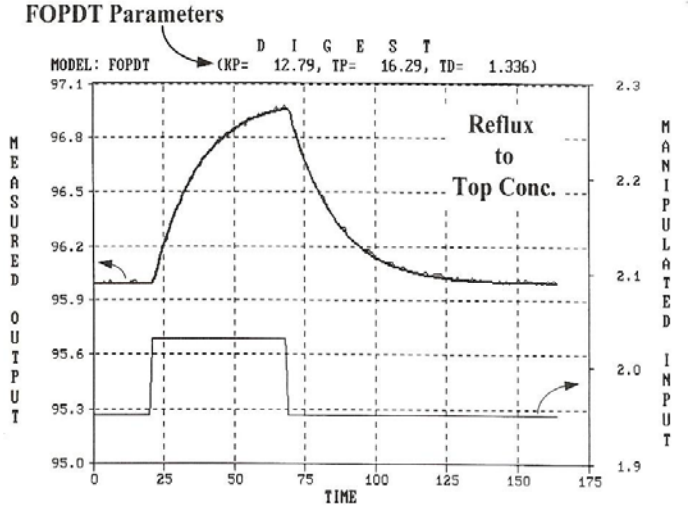


Figure 2 - Fit of FOPDT Dynamic Model to Reflux Rate Pulse Test Data

These dynamic model parameters can now be used in correlations to obtain initial estimates for controller tuning. Digest contains a number of correlations, including IMC (internal model control), Cohen-Coon, IAE (integral of absolute error), and ITAE (integral of time weighted absolute error) and will compute the tuning parameters for a P-Only, PI or PID algorithm at user request. Fig. 3 shows such a computation for the above FOPDT model parameters for a PI controller.

```
TUNING PARAMETERS

Controller Mode: PI

      Internal Model Control (IMC)
TC (Closed Loop Time Constant) =    1.629
Controller Gain =    0.430
Reset Time      =    16.29
      Cohen-Coon
Controller Gain =    0.864
Reset Time      =    3.798
      ITAE (for Set Point Changes)
Controller Gain =    0.453
Reset Time      =    16.03
      ITAE (for Disturbance Changes)
Controller Gain =    0.773
Reset Time      =    4.413
      IAE (for Set Point Changes)
Controller Gain =    0.510
Reset Time      =    16.40
      IAE (for Disturbance Changes)
Controller Gain =    0.906
Reset Time      =    4.573
```

Figure 3 - PI Tuning Parameters Recommended by Digest for the FOPDT Parameters of Fig. 2

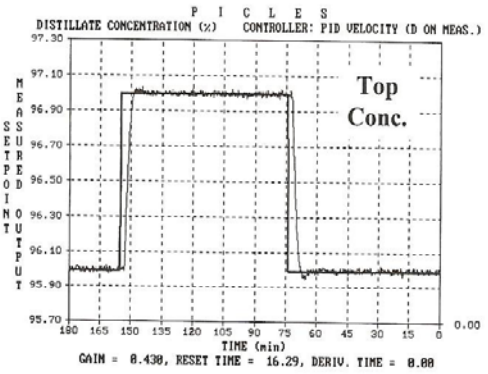
In this case study, IMC controller tuning is used. Circled in Fig. 3 are the IMC tuning parameters computed by Digest for the above FOPDT model parameters. Note that Digest recommends a closed loop time constant following the heuristic that the closed loop time constant, τ_C , equals $0.1\tau_P$ or 0.8θ, which ever is larger. The user can change this closed loop time constant value and Digest will recompute the IMC parameters. Thus, Digest recommends the PI tuning parameters:

$$\text{Controller Gain, } K_C = 0.430 \ \frac{\text{lb / min}}{\%}$$

$$\text{Reset Time, } \tau_I = 16.29 \text{ minutes}$$

Returning to Picles, the Velocity PID with Derivative on Measurement controller is selected and these tuning values are entered. The derivative time is set to zero, resulting in the desired PI controller form. Following the controller design procedure, the controller is now tested to evaluate its performance. Recall the objective of this particular case study is the effective tracking of set point steps for a top concentration in the range of 96% to 97%.

The capability of the top controller to track step changes in distillate concentration set point while the bottom loop is in manual mode is shown in Fig. 4. Although the top controller does an exceptional job of set point tracking, showing a rapid response with minimal overshoot, the corresponding movement in the bottoms concentration shown in Fig. 4 highlights the need for a second controller on the bottoms concentration.



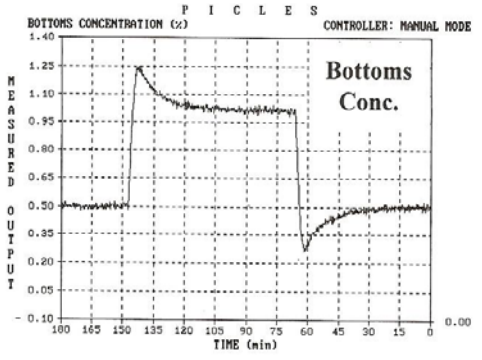


Figure 4 - Set Point Tracking Capability of Top Loop Under PI Control When Bottom Loop is in Manual Mode

Bottoms Control and Loop Interaction

The design of the bottoms PI controller is analogous to that of the top controller. Although plots are not presented for this controller design, the procedure is to pulse the steam rate from its original steady state value of 1.710 lb/min down to 1.685 lb/min. Note that the top controller is in manual mode for this design. These values of the manipulated steam rate variable are chosen as they cause the measured bottom concentration to change in manual mode from a concentration of 0.5% up to the stated objective concentration of 1.0%. After approximately 50 simulation minutes, the steam rate is stepped back up to the original value of 1.710 lb/min. Using Picles' file save facility, the process data is saved to disk using the default sample rate.

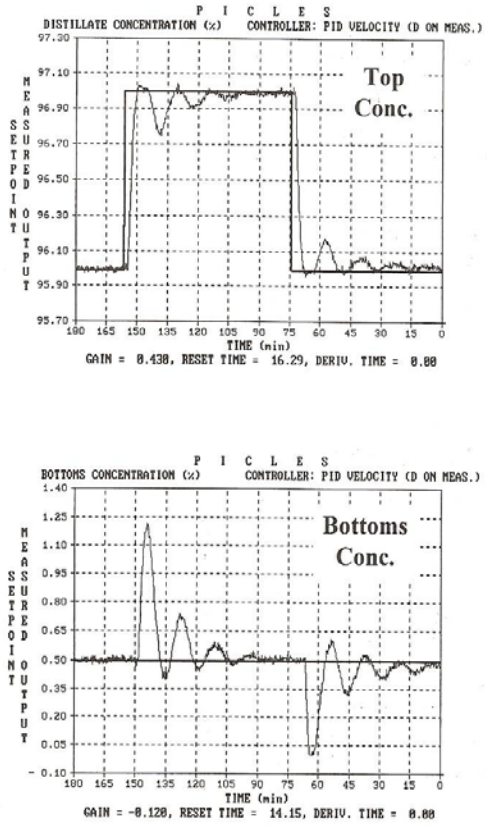


Figure 5 - Set Point Tracking Capability of Top Loop Under PI Control and Interaction with Bottom Loop, Also Under PI Control

Digest computes the FOPDT model parameters for this data as (the subscript BS stands for Bottoms Concentration - Steam Rate):

$$\text{Process Gain, } K_{P,BS} = -19.34 \ \frac{\%}{\text{lb / min}}$$

$$\text{Overall Time Constant, } \tau_{P,BS} = 14.15 \text{ minutes}$$

$$\text{Apparent Dead Time, } \theta_{BS} = 3.392 \text{ minutes}$$

Using these model parameters in the IMC tuning relations and using the default closed loop time constant, Digest recommends the PI controller tuning parameters:

$$\text{Controller Gain, } K_C = -0.120 \ \frac{\text{lb / min}}{\%}$$

$$\text{Reset Time, } \tau_I = 14.15 \text{ minutes}$$

Returning to Picles, the Velocity PID with Derivative on Measurement controller is selected and these tuning values are entered. The derivative time is again set to zero to yield the desired PI controller form. Controller performance in tracking set point steps in bottoms concentration in the range of 0.5% to 1.0% while the top control loop is in manual mode reveal that this controller also does an exceptional job of set point tracking, showing a rapid response with minimal overshoot.

Now, both top and bottom PI controllers are closed using the controller tuning values listed previously. As shown in Fig. 5, the performance of the top controller in set point tracking is investigated. One observation is that the performance of the top controller has degraded somewhat relative to that achieved when the bottom loop was in manual mode (recall Fig. 4). Perhaps more important is that controller interaction causes significant transients in the bottom concentration even though the bottoms concentration set point remains constant.

Decoupler Design

Decouplers are essentially feed forward elements. That is, they are controller elements which counter the interaction that the "cross loop" manipulated variable (a measured disturbance) has on the measured variable of interest.

Consider that changes in steam rate are known to impact the top concentration and that this relationship can be modeled. Using such a model, a decoupler can be designed which manipulates the reflux rate by the proper amount whenever the steam rate changes to counter the impending steam rate impact on top concentration. As shown in Fig. 6, decoupling manipulations must be added to those of the feed back controller to arrive at an overall manipulated variable change.

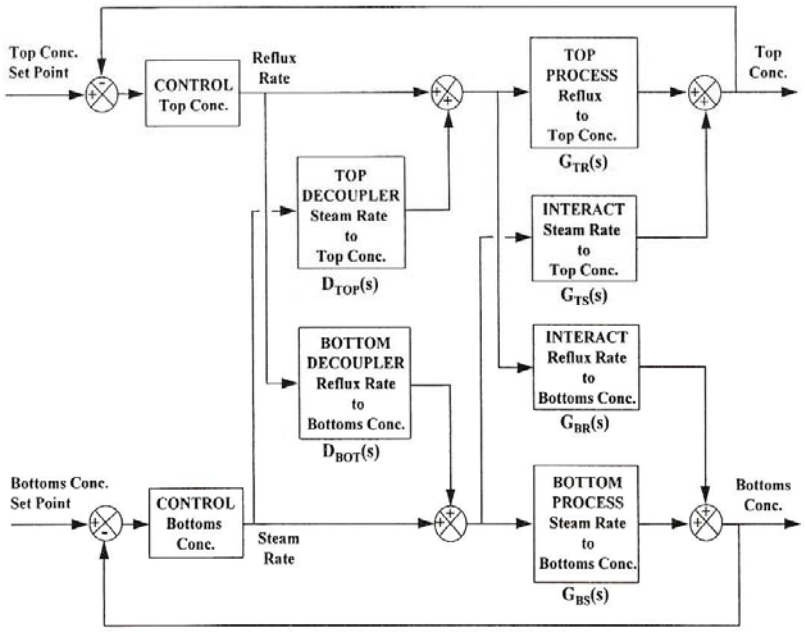


Figure 6 - Block Diagram of Top and Bottom Control Loops Including "Cross Loop" Interaction and Decouplers

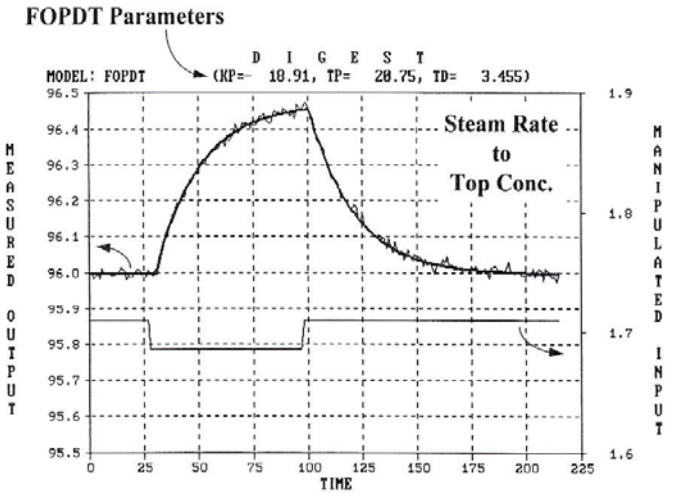


Figure 7 - Fit of FOPDT Dynamic Model to Steam Rate Pulse Test Data With Top Concentration as the Measured Output Variable

In the real world, the blocks labeled $G_{TS}(s)$ and $G_{BR}(s)$ are actual interactions characteristic of the column behavior. For decoupler design, the dynamic behavior of these blocks must be approximated with simple linear models. These models are then used to design the decoupler elements labeled $D_{TOP}(s)$ and $D_{BOT}(s)$.

As detailed in the popular process control text books, the top decoupler, $D_{BOT}(s)$, is defined as:

$$D_{TOP}(s) = -\frac{G_{TS}(s)}{G_{TR}(s)} \qquad (1)$$

If simple FOPDT models are assumed to describe $G_{TR}(s)$ and $G_{TS}(s)$, the reflux rate to top concentration process and steam rate to top concentration cross loop interaction process respectively, then Eqn. 1 can be expressed:

$$D_{TOP}(s) = \left(\frac{K_{P,TS}}{K_{P,TR}}\right)\left(\frac{(\tau_{P,TR})s+1}{(\tau_{P,TS})s+1}\right)e^{(\theta_{TS}-\theta_{TR})s} \qquad (2)$$

Where "s" is the Laplace domain independent variable. Picles takes care of the math, but to design the top decoupler, the user must specify the gain ratio ($K_{P,TS}/K_{P,TR}$), the process time constant ($\tau_{P,TR}$), the cross loop time constant ($\tau_{P,TR}$), and the dead time difference ($\theta_{TS} - \theta_{TR}$).

Recall that $K_{P,TR}$, $\tau_{P,TR}$ and θ_{TR} were computed earlier. No additional step or pulse tests need be performed to compute $G_{TS}(s)$, the steam rate to top concentration FOPDT model, because the data generated from the steam rate to bottoms concentration pulse test already contains the necessary information. Thus, this data set is read into Digest and the steam rate column is labeled as the manipulated input variable as before. Now, however, the top concentration column of data is labeled as the measured output data.

Fig. 7 shows a FOPDT model fit of this data. As shown, Digest computes the model parameters as (the subscript TS stands for Top Concentration - Steam Rate):

Process Gain, $K_{P,TS} = -18.91 \dfrac{\%}{\text{lb / min}}$

Overall Time Constant, $\tau_{P,TS} = 20.75$ minutes

Apparent Dead Time, $\theta_{TS} = 3.455$ minutes

With $G_{TR}(s)$ and $G_{TS}(s)$ now defined, all information required to design $D_{TOP}(s)$, the top decoupler, is available for entry into Picles.

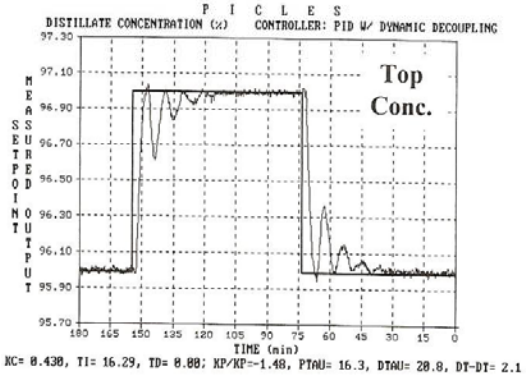
The bottom decoupler design follows an analogous procedure. The reflux rate pulse test data is read into Digest. The reflux rate is labeled as the manipulated input variable as before and the bottoms concentration is labeled as the measured output variable. Digest produces the FOPDT model (the subscript BR stands for Bottom Concentration - Reflux Rate):

Process Gain, $K_{P,BR} = 6.634 \dfrac{\%}{\text{lb / min}}$

Overall Time Constant, $\tau_{P,BR} = 10.67$ minutes

Apparent Dead Time, $\theta_{BR} = 7.205$ minutes

All process models and interaction models are now defined in FOPDT form. Returning to Picles, the Velocity PID with Dynamic Decoupler controllers are chosen for



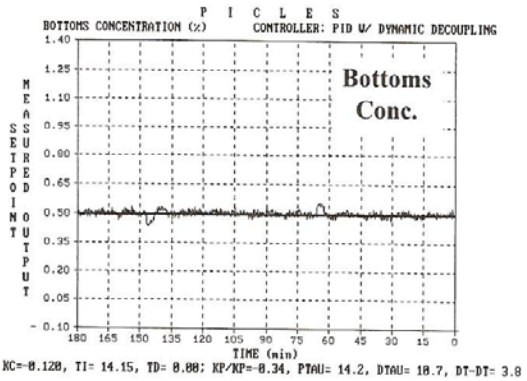


Figure 8 - Set Point Tracking Capability of Top Loop and Reduced Interaction with Bottom Loop When Both are Under PI Control with Dynamic Decoupling

both top and bottom loops. The PI tuning parameters for the feed back loops remain as before. The decoupler model values required by Picles are entered as:

	Top Decoupler	Bottom Decoupler
Gain Ratio	-1.48	-0.34
Process Tau	16.29	14.15
Cross Loop Tau	20.75	10.67
Dead Time Difference	2.12	3.81

Fig. 8 shows the set point tracking capability of the decoupled top concentration PI controller. One observation is that the set point tracking capability of the top PI controller has degraded further because of the decouplers (recall Figs. 4 and 5). The exciting result, however, is that the bottom controller now shows virtually no interaction as the top controller responds to the set point steps.

Homework

Does this end The Case of the Interacting Controllers? Not likely! First, the performance of the bottom decoupler must be investigated. Then, the top and bottom concentration PI controllers must be retuned to improve set point tracking performance. Finally, the sensitivity of decoupler model parameters on decoupler performance must be established.

Other investigations include comparing the performance of static decouplers available in Picles with those of the dynamic decouplers explored in this article, and establishing the benefits of decoupling when full PID control is implemented for the top and bottom feed back loops. The benefits of decoupling when the objective is not set point tracking, but rather, rejecting feed rate disturbances could be investigated. Or a Relative Gain Array analysis could be performed to validate that Picles uses the proper manipulated to measured variable pairing.

Use your imagination to develop a variety of process dynamics and control case studies using Picles and Digest. Needless to say, homework problems which begin, "start with this transfer function and..." sound limiting indeed in comparison to the hands on, real world experiences possible with the Picles training simulator.

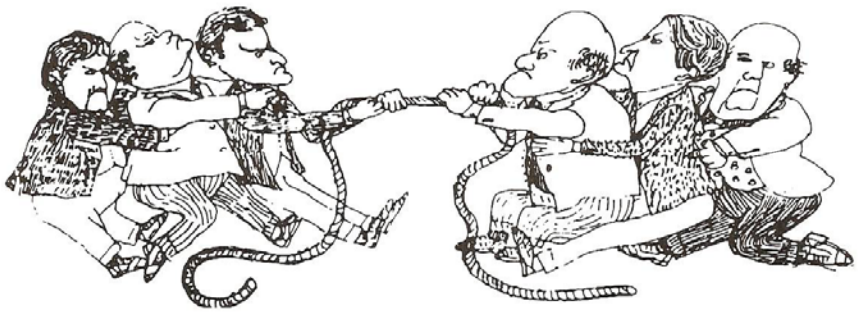
For More Information

For more information about Picles, Digest and available teaching materials, contact:

Doug Cooper
Chemical Engineering Department
University of Connecticut
Storrs CT 06269-3222
Phone: (203) 486-4092
Email: cooper@eng2.uconn.edu

Literature Cited

Wood, R. K. and M. W. Berry, "Terminal Composition Control of a Binary Distillation Column," *Chemical Engineering Science*, **28**, 1707 (1973).



How to choose the right algorithm for a flowsheeting code.

Octave - A High Level Interactive Language for Numerical Computations

By John W. Eaton† and James B. Rawlings††, University of Texas at Austin

Introduction

Octave is an interactive language for numerical computing that is mostly compatible with MATLAB.[1] Originally intended to be companion software for an undergraduate-level textbook on chemical reactor design being written by James B. Rawlings and John G. Ekerdt at the University of Texas, it has become much more than just another "courseware" package with limited utility beyond the classroom. It is currently in use by thousands of people at educational, commercial, and government sites worldwide.

Although our initial goals were somewhat vague, we knew that we wanted to create something that would enable students to solve realistic problems, and that they could use for many things other than chemical reactor design problems. We believe that we are succeeding, as we find students using Octave to solve problems for other courses in which it is not being used as a primary tool for instruction. In our experience, this rarely happened when we emphasized other more traditional languages.

Still, there are those who would say that we should be teaching the students Fortran instead, because that is the language of engineering and scientific programming. But every time we have tried that, the students have spent far too much time trying to figure out why their Fortran code crashes and not enough time learning about chemical engineering. With Octave, most students pick up the basics quickly, and are using it confidently in just a few hours.

In addition to using it as an integral part of teaching reactor design to our undergraduates, Octave is also being used in several other undergraduate and graduate courses in our department, and the math department at the University of Texas has been using it for teaching differential equations and linear algebra as well. We also have reports of it being used for various teaching and research purposes many other places around the world. If you begin using it, please let us know. We are always interested to find out how Octave is being used in other places.

Unlike most of the software that you may be familiar with, everyone is encouraged to share Octave with others under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation (FSF).[2] Briefly, the intent of using the terms of the GPL is to ensure that everyone has the ability and the right to copy, redistribute, understand, and modify the software, and to make sure that no one takes these rights away from you (and also that you do not take these rights away from other people). The complete text of the GPL is available with the Octave source and binary distributions, and via anonymous ftp from `prep.ai.mit.edu` in the file `/pub/gnu/COPYING`. The complete source code for Octave is freely available via anonymous ftp (see page 16 for instructions for how obtain your own copy). You are also encouraged to help make Octave more useful by writing and contributing additional functions for it, and by reporting any problems you may have.

Overview of Octave

Octave is primarily a matrix language, but it does provide some facilities for managing data structures and handling strings. There are built-in functions for solving linear and nonlinear equations, integrating functions on finite and infinite intervals, and solving sets of ordinary differential and differential-algebraic equations.

The Octave interpreter is written in a mixture of C and C++, but most of the numerical methods are handled by standard Fortran libraries such as the BLAS [7,3,2],

† jwe@che.utexas.edu

†† jbraw@che.utexas.edu

1 MATLAB is a trademark of the Mathworks Inc.

2 Founded in 1985 by Richard Stallman, the FSF is a non-profit organization dedicated to eliminating restrictions on people's abilities and rights to copy, redistribute, understand, and modify computer programs. It is important to note that the word "free" in the name refers to freedom, not price. The FSF is currently at work on a freely redistributable software system named GNU that will be upwardly compatible with Unix. Most parts of this system are already working, and are being distributed now. For more information, contact the FSF at the address given on page 16.

LAPACK [1], MINPACK [8], QUADPACK [10], ODEPACK [6], and DASSL [9]. To smoothly interface with the interpreter, the numerical libraries have been packaged in a library of C++ classes. If possible, the Fortran subroutines are compiled with the system's Fortran compiler, and called directly from the C++ functions. If that's not possible, you can still compile Octave if you have the freely available Fortran to C translator `f2c`.

Octave can also solve nonlinear programming problems using NPSOL [5] and QPSOL [4], but since these subroutine packages are not freely redistributable, we are unable to include them with the standard Octave distribution. We would like for all users of Octave to be able to easily solve nonlinear programming problems, however, and would be interested in learning of freely redistributable alternatives.

If you are unfamiliar with matrix languages like MATLAB, Xmath, or MatrixX, the following examples will probably help to give you a better feel for the possibilities than any description we could write.

Creating a matrix. To create a new matrix and store it in a variable so that it you can refer to it later, you can use a command like

```
octave:1> a = [1, 1, 2; 3, 5, 8; 13, 21, 34]
```

where commas are used to separate elements on the same row, and semicolons are used to separate rows. Octave will respond by printing the matrix in neatly aligned columns. Ending a command with a semicolon tells Octave to not print the result of a command. For example,

```
octave:2> b = rand (3, 2);
```

will create a 3 row, 2 column matrix with each element set to a random value.

To display the value of any variable, you simply type the name of the variable. For example, the command

```
octave:3> b
```

displays the value stored in the matrix `b`.

Matrix arithmetic. Octave has a convenient operator notation for performing matrix arithmetic. For example, the command

```
octave:4> 2 * a
```

multiplies the matrix `a` by the scalar 2, the command

```
octave:5> a * b
```

multiplies the matrices `a` and `b`, and the command

```
octave:6> a' * a
```

forms the matrix product $A^T A$. In each case, if the expression does not end in a semicolon, Octave will respond by displaying the result of the calculation.

Solving linear equations. The expression

```
octave:7> a\b
```

solves the set of linear equations $Ax = b$. This is conceptually equivalent to $A^{-1}b$, but avoids computing the inverse of the matrix directly. If the coefficient matrix is singular, Octave will print a warning message and compute a minimum norm solution.

Integrating differential equations. Octave has built-in functions for solving nonlinear differential equations of the form

$$\frac{dx}{dt} = f(x,t), \qquad x(t-t_0) = x_0$$

```
octave:8> function xdot = oregonator (x, t)
>
>   if (nargin != 2)
>      usage ("oregonator (x, t)");
>   endif
>
>   xdot = zeros (3,1);
>
>   xdot(1) = 77.27 * (x(2) - x(1)*x(2) + x(1) - 8.375e-6*x(1)^2);
>   xdot(2) = (x(3) - x(1)*x(2) - x(2))/ 77.27;
>   xdot(3) = 0.161*(x(1) - x(3));
>
>   endfunction
```

Figure 1. Function defining a set of differential equations.

For Octave to integrate equations of this form, you must first provide a definition of the function $f(x,t)$. This is straightforward, and may be accomplished by entering the function body directly on the command line. For example, the function shown in Figure 1 defines the right hand side function for an interesting pair of nonlinear differential equations. Note that while you are entering a function, Octave responds with a different prompt, to indicate that it is waiting for you to complete your input.

Given the initial condition

```
x0 = [4.0, 1.1, 4.0];
```

and the set of desired output times as a column vector `t`, it is easy to integrate the set of differential equations:

```
x = lsode ("oregonator", x0, t);
```

The function `lsode` uses the function lsode from the ODEPACK library. There is also a similar function for solving differential-algebraic equations of the form

$$g(dx/dt, x, t) = 0, \qquad x(t=0) = x_0$$

Producing graphical output. To display the solution of the previous example graphically, use the command

```
plot (t, x)
```

If you are using the X Window System, Octave will automatically create a separate window to display the plot, shown in Figure 2. If you are using a terminal that supports some other graphics commands, you will need to tell Octave what kind of terminal you have. Octave uses `gnuplot`[3] to display graphics, and can display graphics on any terminal that is supported by `gnuplot`.

We have also shown only a small fraction of the features available in Octave. There are currently more than 400 functions included with the standard Octave distribution, and many more will appear in future releases.

Design Goals and History

When we first discussed the possibility of writing some software to supplement the chemical reactor design text, we considered writing some specialized modules for solving problems specific to reaction engineering. It was hard to get excited about yet another subroutine library (with or without snappy graphics) that would only be useful for a narrow range of academic problems. We eventually realized we would be much more effective in getting the students to use the software if we were also excited about using it. We needed something that we would be able to use for research as well as for teaching.

To be effective as a tool for research *and* teaching, we needed to design something that could be easily extended. Users would have to be able to add functionality to be able to solve new and interesting problems, preferably without having to do any low-level programming. This meant we would need to provide users with a relatively high-level mathematical programming language.

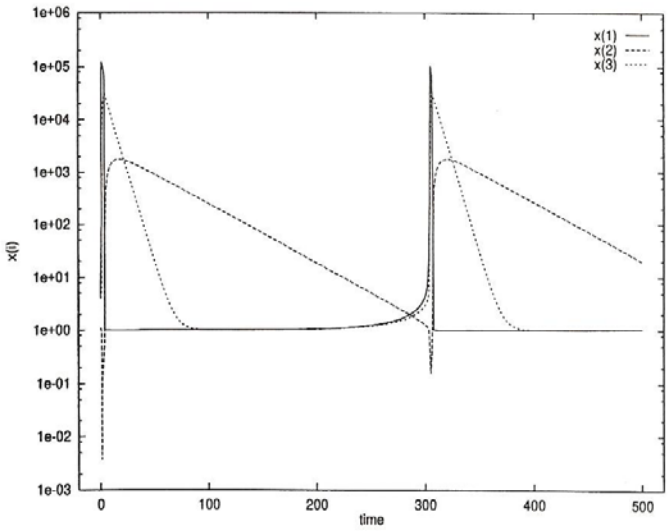


Figure 2. Graphical solution of the system of equations defined by the `oregonator()` function.

3 Curiously, `gnuplot` was neither written nor named for the GNU project; the name is a coincidence.

Rather than invent a completely new language, we chose to implement something that would be mostly compatible with MATLAB. Many engineers and scientists already know and use the language, and most are reluctant to invest the time necessary to learn yet another specialized language. Currently, many people are able to take functions that they have written for MATLAB and use them with Octave without having to make any changes.

Though Octave is compatible with MATLAB in many ways, it is intended to be more than just a clone. Octave adds many interesting new features and extends the language in fundamentally new ways. Furthermore, because it is available in source form, anyone can experiment with adding new features or modifying the language.[4]

The design, organization, and development of Octave is primarily the work of the first author. However, significant portions of Octave were written by others. This includes most of the numerical libraries, the command line editing facility, and the on-line manual browser. Because we were able to take advantage of quite a bit of freely available software, Octave has developed into a substantial software system in just over $2\frac{1}{2}$ years.

In January 1993, about seven months after full time development began, the first public alpha-test release of Octave was announced to the Usenet newsgroup `sci.math.num-analysis`. A small number of brave individuals were curious enough to give it a try, many of them providing a great deal of help working out early bugs in Octave's design and implementation. A little over a year later, in February 1994, the first public release of 1.0 was announced to several Usenet newsgroups, and the FSF began distributing it on tape. In the eleven months that version 1.0 was available, more than 2700 different sites copied the software from our ftp site alone. Since everyone is free to redistribute Octave, we have no way of knowing exactly how many copies have been distributed. In the first ten days that version 1.1 was available, Octave made its way to more than 300 sites around the world.

Features

Octave also provides a number of features that go well beyond what is capable with MATLAB, including data structures with a C-like syntax, the ability to pass any number of parameters to and return any number of values from a function, the ability to define functions directly on the command line, and the ability to define multiple functions in a single input file.

Data structures. One of the MATLAB language's greatest strengths is its ability to easily manipulate matrices. The choice of a matrix of values as the basic data type for a numerical language is a good one, but the fact that it is the *only* data type is severely limiting. Octave's data structure type makes it easy to organize collections of related data and treat them as a single object. For example, the statements

```
x.a = [1, 2, 3, 4];
x.b = "Hello, world!";
```

create a data structure with two elements, `a` and `b`. The variable `x` may then be copied or passed to functions, with the effect of passing all the data referred to by the elements of the structure.

Variable number of arguments for functions. Although it is possible for MATLAB functions to accept different numbers of arguments on different calls, the implementation of a function must place an absolute upper bound on the number of arguments that it can accept. Octave does not impose any limit on the number of parameters that a function can accept. For example, in MATLAB

```
function f (p1, p2, p3, p4, p5, p6)
% XXX add more parameters named pNUM if needed
for i = 1:nargin
   eval (['disp (p', num2str (i), ')']);
end
```

is a typical way to write a function that must handle a variable number of parameters. If this function is called with more than 6 paramters, it fails. To fix the problem, one must edit the function definition to allow more parameters to be passed. The equivalent function in Octave would be written as

```
function f (...)
for i = 1:nargin
disp (va_arg ());
end
```

where the function `va_arg()` returns the next value passed to the function each time it is called. The syntax for this facility is borrowed from the C programming language. A similar mechanism is available to avoid placing an upper limit on the number of outputs a function may produce.

Multiple functions in a single file. With Octave, it is possible to write complete programs including many functions in a single file. This is often convenient for simple programs that need multiple functions to help

[4] It is true that user-defined functions allow a significant level of extensibility, but not everything can be done in an M-file.

program clarity and organization, but that would best remain in a single file for editing purposes.

The only requirement is that a function must be defined in order to be evaluated. However, this does not prevent the following from working

```
function f ()
   g ();
endfunction
function g ();
   printf ("Elvis has left the building.\n");
end
f ();
```

because it is only necessary for the definition of the function `g()` to be available when the function `f()` is *called*. While `f()` is being defined, Octave simply notes that the function `g()` will be called but it does not attempt to find the definition of `g()` until it is required.

Functions can be defined on the command line. In addition to allowing multiple functions to be defined in a single file, Octave allows you to compose functions directly on the command line. This is sometimes convenient for one-shot programs or for solving simple examples without having to edit external files.

Range checking for arrays. Sometimes it is convenient to be able to resize arrays simply by assigning a new value in a previously undefined element. For example, given the array `a = [1,2,3]`, MATLAB allows the statement `a(5) = 5` to change the value of the array `a` to be `[1,2,3,0,5]`. This automatic resizing is not always desirable, however, since an assignment outside the current bounds of an array may indicate an error in the design of an algorithm. If you do not intend to make use of the automatic resizing feature, you may prefer to receive an error message instead. Octave allows you to select the behavior you desire by providing the built-in variable `resize_on_range_error`. If this variable has the value `"false"`, Octave will signal an error instead of allowing such assignments to automatically resize a matrix.

In addition to `resize_on_range_error`, Octave also includes a number of other built-in variables that can be used to control its behavior. You may change the values of these built-in variables at any time, so it is possible to customize Octave's behavior as it is running.

Command completion, editing, and history. In addition to allowing you to edit and store previously entered commands, Octave can complete the commands you type, making long variable and function names much more convenient to use. To ask Octave to complete a command or variable name after typing the first part yourself, you simply type a `TAB` character. If a unique completion exists, Octave will insert the remaining characters for you. If more than one possible completion exists, Octave will insert as much as it can. If multiple completions still exist, typing `TAB` twice will cause Octave to present a list of possibilities. Completion works for language keywords and the names of variables, functions, and files.

Documentation. Octave's 230 page manual includes descriptions of most functions as well as some tutorial material to help get new users started. We use the manual as an optional text in the courses we teach with Octave.

The manual is written using Texinfo, the GNU project's documentation system that uses a single source file to produce both on-line information and printed output. This means that the complete text of Octave's manual is available on line and in printed form.

The on-line form of the manual is often preferred, since it allows convenient methods for searching and following cross references. Users can access the online manual from the shell, or directly from the Octave prompt, using the `help` command. It is also possible to format the on-line manual with hypertext markup language (HTML) commands and read it with a World Wide Web (WWW) browser such as Mosaic.[5]

In addition to allowing you to browse the manual, Octave's help command can also extract and display the first block of comments from functions that you write, so that you can easily provide on-line documentation for them.

Current Educational Uses

Four different professors have used Octave successfully in three undergraduate and three graduate level courses in our department. The courses for undergraduates include Transport Phenomena, Chemical Reactor Analisys and Design, Computer Applications in Chemical Engineering, and the graduate level courses include Advanced Analysis for Chemical Engineers (an applied mathematics course for first year graduate students), Fluid Flow and Heat Transfer, and Elements of Modern Control Theory. Together with the symbolic manipulation package Maple, Octave has also been used for teaching linear algebra and differential equations in the math department at the University of Texas.

Students from four research groups in our department are quite active users, solving problems varying from

5 Open the URL http://www.che.utexas.edu/cgi-bin/info2www?(octave) to see this in action.

model predictive control to kinetic parameter estimation as part of their Ph.D. research projects. A number of students have also used the numerical functions from Octave's C++ library directly. They are typically able to reduce the amount of time spent on code development, and find that the C++ interface to the standard numerical libraries is a convenient alternative to programming in Fortran.

Outside the Univeristy of Texas we have reports of Octave being used for teaching courses in econometrics, time series analysis, and neural networks for applied mathematics and computer science students, and for teaching a course in multivariate data analysis for senior level biology students. A number of groups are also actively using Octave in their research.

Availability

If you are on the Internet, you can copy the latest distribution version of Octave from the file `/pub/octave/octave-M.N.tar.gz`, on the host `ftp.che.utexas.edu`. This tar file has been compressed with GNU `gzip` (also available in the directory `/pub/gnu`), so be sure to use binary mode for the transfer. `M` and `N` stand for version numbers; look at a listing of the directory through ftp to see what version is available. After you unpack the distribution, be sure to look at the files `README` and `INSTALL`.

Octave has been developed with Unix systems in mind, and should work on just about any Unix system that has a working port of the GNU C and C++ compilers. A volunteer is currently working on porting it to DOS and OS/2 systems.

Along with the the source code, binaries for several popular systems are also available. Because we can only make binaries for a few types of systems and must depend on volunteers to help out with the rest, we may not have the latest version available for less popular machines.[6] If you would like to help by building binaries, please contact `bug-octave@che.utexas.edu`.

The source code for Octave is also distributed by the FSF on CD-ROM and tape, and by anonymous ftp from the host `prep.ai.mit.edu` in the directory `/pub/gnu` (the contents of this directory are also mirrored on many other systems around the world, so you may be able to find an ftp site that is much closer to you).

For information about ordering Octave on tape or CD from the FSF, contact `gnu@prep.ai.mit.edu` or write to the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.

Support

Octave is actively maintained by the first author, who is currently busy working on version 1.2 (see below for a list of likely features to be included in future versions). Although we cannot guarantee responses to all bug reports and questions, we are interested in making Octave more reliable, and most bug reports are handled within a matter of days.

An electronic mailing list with approximately 150 subscribers exists for the discussion of issues related to installing, using, and developing Octave. There is not too much traffic yet, though there has been a steady increase since the release of version 1.1. Contact `help-octave-request@che.utexas.edu` if you are interested in joining the discussion. An archive of old messages sent to the list is available on `ftp.che.utexas.edu`.

Because the complete source code for Octave is distributed under the terms of the GPL, Octave will always be freely available, and users will have the option to support and modify the code on their own, or by hiring others to do so. By using Octave, you are not limited to seeking support from any one organization or individual.

Future

In a relatively short period of time, Octave has become a quite capable system for solving many numerical problems. Still, there is always room for some improvement in any software system, and Octave is no exception. Some of the features that are likely to change for the next release include plotting, dynamic linking of externally compiled functions, and full support for DOS and OS/2. Some long-term goals include adding a programmable graphical graphical user interface, improving the overall efficiency of the language, and automatic generation of C++ code.

Improved plotting. It was reasonably easy to incorporate support for `gnuplot` into Octave, but there is substantial room for improvement, particularly in three-dimensional graphics.

Improved dynamic linking. Octave is currently capable of dynamically loading and executing functions written in C++ or other compiled languages, but this only works on systems that have a working version of the GNU dynamic linker, `dld`. We would like to make this feature work on more systems, either by porting `dld` to additional systems, or by using system-specific dynamic linking facilities.

6 We currently have binaries DEC Alpha systems running OSF/1, DECstations running Ultrix, HP Apollo and Motorola 68000 systems running HP/UX, IBM RS/6000 systems running AIX, Intel 386, 486, and Pentium systems running Linux, FreeBSD, or NeXTStep, SGI systems running IRIX, and Sun SPARC systems running SunOS 4.1.x or Solaris 2.

Full support for DOS and OS/2. It is possible to build a binary that will run on OS/2 and DOS systems using the EMX port of the GNU C compiler, but as of this writing, it is not ready for general use. We hope that by the time you are reading this, a fully functional port of Octave will be available for these systems.

Programmable GUI. Many people would like to see Octave have the ability to display windows with pushbuttons, scrollbars, and sliders, particularly if they can be integrated with built-in plotting capabilities. We agree, though we don't place as high a priority on this as we do on Octave's numerical functions.

Improved efficiency. Most research problems that we solve, and most problems that we assign in the courses that we teach, are relatively small, and we usually write prototype solutions for them. For these applications, Octave's interpreter is sufficiently fast, so efficiency is a generally a minor concern. Octave's primary advantage is its ease of use. For solving larger problems, or developing code for a production system that will be executed thousands of times, an interpreted language like Octave is probably not the best long-term solution. However, even for small problems, some operations are notably slower than others and could use a boost in performance. We are always looking for opportunities to improve the speed of the interpreter provided that the effort required is within reason.

Code generation. It is relatively easy to develop code in Octave, but for many operations, even the best interpreter will be significantly slower than compiled C++ or Fortran. If Octave were capable of generating C++ code, it would be possible to take advantage of Octave as a convenient development environment to produce reasonably fast solutions to larger problems.

Acknowledgements

Many people have already contributed to Octave's development by writing code, submitting bug reports, and suggesting various improvements. Several of the more significant contributions include a collection of statistical, elementary, and special functions from Kurt Hornik and his colleagues at the Technical University of Vienna Department of Probability and Statistics, a number of linear algebra and linear control systems functions from Scottedward Hodel and Bruce Tenison at the Auburn University Department of Electrical Engineering, and an improved set of plotting functions from Rick Niles at NASA's Goddard Space Flight Center. A complete list of the individuals who have contributed appears in the Octave manual, and although we do not have room to list them all here, we are grateful for their continuing help.

Special thanks are in order to the following people and organizations who have supported us along the way.

- Digital Equipment Corporation, for an equipment grant as part of their External Research Program.
- Sun Microsystems, Inc., for an Academic Equipment grant.
- Noel Bell and Texaco Chemical Company, for providing funding necessary to continue the development of Octave.
- The University of Texas College of Engineering, for providing a Challenge for Excellence Research Supplement, and for providing an Academic Development Funds grant.
- The State of Texas, for providing funding through the Texas Advanced Technology Program under Grant No. 003658-078.
- Richard Stallman, for writing GNU.

This project would not have been possible without the GNU software used in and used to produce Octave.

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen. LAPACK Users' Guide. SIAM, Philadelphia, 1992.

[2] Jack J. Dongarra, Jeremey Du Croz, Sven Hammarling, and Iain Duff. A Set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Soft.*, 16(1):1-17, 1990.

[3] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard J. Hanson. An Extended Set of FORTRAN Basic Linear Algebra Subprograms. *ACM Trans. Math. Soft.*, 14(1):1-17, March 1988.

[4] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's Guide for SOL/QPSOL: A Fortran Package for Quadratic Programming, technical report SOL 83-12. Technical report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1983.

[5] Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright. User's Guide for SOL/NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming, technical report SOL 86-2. Technical report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1986.

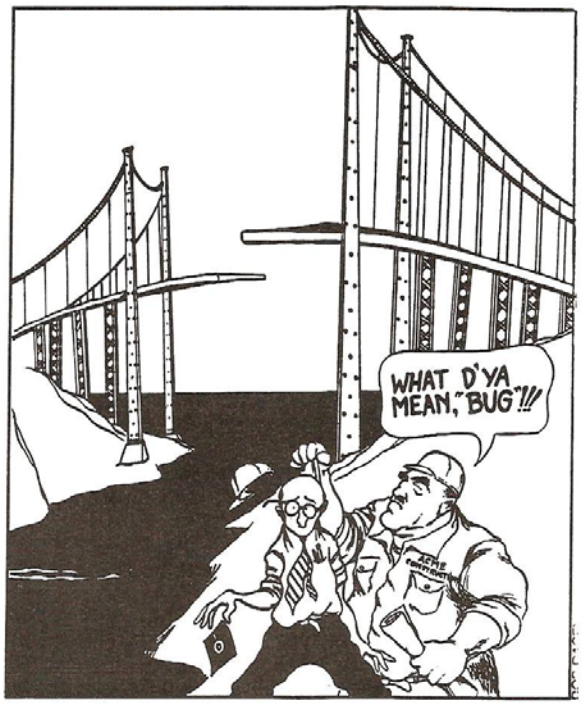
[6] Alan C. Hindmarsh. ODEPACK, Systematized Collection of ODE Solvers. In R. S. Stepleman, editor, *Scientific Computing*, pages 55-64, Amsterdam, 1983. North-Holland .

[7] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for Fortran Usage. *ACM Trans. Math. Soft.*, 5(3):308323, September 1979.

[8] Jorge J. More, Burton S. Garbow, and Kenneth E. Hillstrom. User Guide for Minpack-1. Technical Report ANL-80-74, Argonne National Laboratory, March 1980.

[9] L. R. Petzold. A description of DASSL: A Differential-Algebraic System Solver. In R. S. Stepleman, editor, *Scientific Computing*, pages 65-68, Amsterdam, 1983. North-Holland.

[10] Robert Piessens, Elise de Doncker-Kapenga, Christoph W. Uberhuber, and David K. Kahaner. *QUADPACK: A Subroutine Package for Automatic Integration.* Springer-Verlag, Berlin, 1983.



Datamation 1986

Monsanto's FLOWTRAN System: A Historical Perspective

By Edward M. Rosen, EMR Technology Group, Monsanto Company (Retired)

Introduction

Monsanto's FLOWTRAN (FLOWsheet TRANslator) system has been widely recognized as the grandfather of modern process simulation systems. It has had a broad influence on undergraduate chemical engineering design education and on the commercial practice over a span of some 25 years. The year 1994 generally brings its life cycle, which began in the mid 1960's, to a close.

The FLOWTRAN System

FLOWTRAN is a block oriented process simulation system. Each block (model) generally represents a plant processing step in which output streams are calculated from input streams subject to block parameters. The term sequential-modular is generally used to describe its architecture (Motard et al., 1975). Throughout, it is coded in FORTRAN.

The system grew to include three major components:

1. The FLOWTRAN simulator
2. PROPTY - the pure component physical property processing program that fits raw pure component data to FLOWTRAN physical property models.
3. VLE - The program that fits mixture vapor-liquid equilibrium data to FLOWTRAN activity coefficient models.

Generally the system was designed for chemical vapor-liquid processes. It did not handle solids processes.

As the system evolved in response to changes in computer technology and was spun off to serve different roles, modifications were made to each of its versions. It was used:

1. As the process simulation system within Monsanto.
2. As a commercial time-sharing service (Computerized Engineering Business)
3. As a licensed product (FLOWTRAN Licensing)
4. As an aid to industry (FLOWTRAN/MIT project)
5. As an aid to education (CACHE/FLOWTRAN project)

FLOWTRAN in each of these roles is described below.

Early Background

In the late 1950's a number of events occurred which, in retrospect, laid the foundation for the creation of FLOWTRAN (Rosen and Kaufman, 1981). In 1957 FORTRAN was developed. Monsanto tested it out on a cooler-condenser program on McDonnell Douglas's IBM 704 (Pauls, 1993) and obtained an IBM 704 itself in March 1959. It was the first such machine installed by a chemical company.

When the author arrived in September of that year, a small Applied Mathematics group headed by Leon Cooper was in place in Monsanto's Corporate Research Department and was beginning to apply the new digital computer technology to company problems. (Leon Cooper later left Monsanto to found the Applied Mathematics Department at Washington University. Subsequently he was a dean at Southern Methodist University). The directions for research using this new technology were wide open.

The application of the digital computer to the simulation of chemical processes was just emerging. The work of Nagiev became the basis of a material balance program (Rosen, 1962) which had only limited appeal to the engineers in Monsanto, however. Though suitable for processes which had a large number of recycle loops, it depended on the solution of large sets of linear equations and was difficult for engineers to understand unless the overall solution converged. (Later the basic solution architecture became known as simultaneous-modular (Motard et al., 1975; Rosen, 1980)). To force solution to the split fractions of each building block, Wegstein's method (Wegstein, 1958) was used. A modified and bounded form of this method later became widely used in process simulation. In early 1964, R. H. Cavett, then at Pure Oil Co. initiated a correspondence with the author (Rosen, 1964) and later that year he joined the Applied Mathematics group at Monsanto under the direction of J. B. Duncan. The group consisted of H. J. Morris, Group leader, R. H. Cavett, C. W. DiBella, J. Sinai, E. M. Rosen and P. B. Zwart as well as W. E. Ball of Washington University. In March of 1966, L. Bellamy joined the group.

At Pure Oil, Cavett had carried out much of the early physical property and numerical analysis work that was

to become part of FLOWTRAN (Cavett, 1962, 1963, 1964a,b,c,d). He later suggested the name FLOWTRAN for the system he had in mind.

Monsanto purchased all rights to the Pure Oil work. The work reflected Bob's conviction that theoretical and semi-theoretical physical property equation models should be incorporated into the simulation (rather than empirical curve fits), despite the fact that with the technology of the time, the time penalty paid for the computations was substantial. (This issue, in fact, has persisted over the years). The work also suggested, for the first time, the concept of the convergence block, in which recycle loop convergence could be forced in an independent building block.

The sequential-modular approach (advocated by Bob Cavett) vs. the simultaneous-modular approach (advocated by the author) for the architecture of the proposed simulator gave rise to considerable debate. Both approaches depended on the building block concept but the sequential-modular formulation was more appealing to the engineer. With the numerical methods available at the time, the sequential-modular system was also more efficient on the test problems tried.

The idea of incorporating a nonlinear optimizer to manipulate the building block parameters to maximize a specified objective function was very much a part of the overall system planning. However, the limitations of the hardware and the marginal interest of engineers in searching for an optimal solution rather than simply a feasible solution made this option a low priority item. (A number of years later an optimizer was added to the University version of FLOWTRAN (Parker, 1979; Parker and Hughes, 1981a,b; Biegler and Hughes, 1981; Lang and Biegler, 1987)).

Initially FLOWTRAN required the coding of a main FORTRAN program consisting of a series of CALLS to FORTRAN coded building blocks. Nevertheless, the system was released for use in Monsanto on April 1, 1966 and a preprocessor, which wrote the main FORTRAN program, was added later that year by Bill Ball.

Authors of the early building blocks included R. H. Cavett, C. W. DiBella, H. J. Morris, J. Sinai, and J. B. Duncan. A separate FLOWTRAN application group was formed in the Central Engineering Department under Paul E. Perisot in 1966 with H. H. Chien, C. B. Cobb, J. T. Sommerfeld and Chuck Hayward.

Early applications were carried out by H. J. Morris, J. Sinai, R. H. Cavett, C. W. DiBella, L. Bellamy and B. T. Fairchild. Additional applications were carried out by H. H. Chien, J. T. Sommerfeld and C. B. Cobb (Chien et al., 1969).

The author moved to the Corporate Management Information and Systems Department (MISD) at Monsanto in April 1966. FLOWTRAN followed with L. Bellamy and C. W. DiBella in January, 1967. Cavett and Morris remained in the Central Research Department.

FLOWTRAN's development continued in MISD in the late 1960's. A large number of people contributed to FLOWTRAN during the this time. These included J. R. Deam, R. S. Frank, A. H. Larsen, D. J. Nitsche, K. G. Pollack and G. F. Schoditsch in MISD and H. H. Chien and J. T. Sommerfeld in the Corporate Engineering Department. In MISD the system was managed in turn by D. J. Kaufman, V. P. Herbert, C. W. DiBella and E. M. Rosen. In April of 1972, FLOWTRAN was transferred to the Corporate Engineering Dept. Successively, the system maintenance was the responsibility of A. C. Pauls, S. I. Proctor, L. C. McCune, and J. M. Hoffmann while system development and user support was the responsibility of H. H. Chien with A. H. Larsen, D. R. Schneider, R. V. Sanderson, P. D. Shoemaker, S. Sanders and B. A. Ross and then C. W. DiBella. System and development responsibilities were consolidated in 1977 under D. A. Novak and since continued under A. H. Larsen and H. H. Chien. (Chien, 1972a,b, 1973, 1978; Sanderson and Chien, 1973; Chien and Larsen, 1974; Larsen, 1977, 1982; Rosen and Pauls, 1977).

The author's book (Henley and Rosen, 1969) was largely inspired by FLOWTRAN. The book included unpublished work on the application of Quasi-Newton methods to recycle calculations and an algorithm for three-phase flash calculations (Seader, 1989).

Bob Cavett suffered a massive cerebral hemorrhage on January 12, 1976. He slowly recovered and was named the first recipient of AIChE CAST Division's Computing Practice Award in 1987. He recently completed his PhD at Arizona State University (Cavett, 1993).

Computerized Engineering Business

In 1968, the FLOWTRAN system (which internally was running on an IBM 360/65) was converted for use on CDC 6000 series computers so the system could be made commercially available by service bureau access through United Computing Services (C&EN Staff, 1970; Chemical Week Staff, 1970; Rorschach, R. L. et al., 1970; Harris, 1971). R. H. Cavett was manager of operations and systems maintenance and Henry J. Morris was commercial development manager in the newly formed Computerized Engineering effort in the New Enterprise Division of Monsanto (Cavett, 1970, 1972a,b).

The system was test marketed in early 1969 and the first outside customer was signed in July 1969. By the end of the test market period at year end, there were eight customers and by the second quarter of 1970, the users had tripled to include major companies producing organic chemicals, petrochemicals, oil and gas, and contract engineering and consulting firms.

Responsibility for the Computerized Engineering effort was transferred to Monsanto Enviro-Chem Systems in 1971 and by 1973 some 70 companies from over 200 locations had signed contracts with the FLOWTRAN service bureau. However, in January 1973, Enviro-Chem decided to cancel this service.

FLOWTRAN Licensing

The cancellation of the commercial service bureau business caused customers to request that FLOWTRAN be licensed to run on their own machines.

This was done and a FLOWTRAN Users Group was formed in January, 1977 to share experiences and FLOWTRAN enhancements. The licensing business was conducted by the Corporate Engineering Department. By 1983 some 10 customers had obtained licenses. (Monsanto Enviro-Chem Systems, 1972).

The Monsanto/MIT-ERDA (ASPEN) Project

During much of 1977 MIT carried on a discussion with Monsanto to utilize the source code of the external version of FLOWTRAN as a basis for the development of a new process simulator.

The new simulator was especially concerned with a capability to handle solids so as to be able to simulate many of the coal-based energy projects under study in a number of government and industrial laboratories. A contract was signed in January of 1978 and H. H. Chien of Monsanto joined the System Design Task Force of the ASPEN (Advanced System for Process Engineering) Project Advisory Committee composed of a broad representation of industry. Professor Larry Evans of MIT and Professor Warren Seider (University of Pennsylvania) directed the three year project, budgeted at $3,285,000 by DOE. Dr. Paul W. Gallier was loaned by Monsanto as the project manager. A payment was made to Monsanto for the source license.

Testing of the ASPEN system (Phase II) began in September 1979 (to September, 1981) at a number of industrial companies. ASPEN was subsequently made publicly available.

The Monsanto/CACHE FLOWTRAN Project

The CACHE (Computer Aids for Chemical Engineering Education) Committee was formed in April 1969 at a meeting in Ann Arbor, Michigan of 14 educators. The principal goal of the Committee was to "accelerate the integration of digital computation into chemical engineering by inter-university cooperation in the preparation of recommendations for curriculum and course outlines, and the development of new computing systems." Initially CACHE was sponsored by the National Academy of Engineering. In 1975 it was formally organized as the CACHE Corporation, a not-for-profit entity.

During 1972, the Large Scale Task Force of the CACHE Committee under the direction of J. D. Seader and W. D. Seider, Chairman of the Program Distribution Task Force, completed an evaluation of industrial computer-aided steady state process design and simulation programs that might be made available for academic use. The most flexible and sophisticated program found was the FLOWTRAN system of Monsanto. CACHE approached Monsanto through the offices of J. J. McKetta of the University of Texas and F. E. Reese of Monsanto as to whether FLOWTRAN could be made available for educational purposes to the universities. As a result Monsanto granted this request (Dec 10, 1973) and made the system available on Remote Job Entry or Remote Batch basis through United Computing Services (Kansas City) in 1974 (AIChE Staff, 1974). A. C. Pauls was assigned by Monsanto as a consultant to CACHE to assist in the use and promotion of the system.

By August 1974, he, J. D. Seader and W. D. Seider completed a 375 page textbook on the use of FLOWTRAN, entitled "FLOWTRAN Simulation - An Introduction", which was published at low cost by CACHE. This textbook included a chapter by R. R. Hughes on a comprehensive example to illustrate the application of many of the features of FLOWTRAN.

Also in August, 1974, the first of three workshops on the use of FLOWTRAN for academics was held at Northwestern University. It was attended by 38 faculty members representing 35 different schools in the United States and Canada. Two additional workshops were held in Houston and Boston in 1975. Early in 1975, R. R. Hughes completed a 56 page guide to the use of FLOWTRAN on the UCS system. By mid-1975, 25 schools were using FLOWTRAN in teaching and research, and J. T. Sommerfeld and J. P. Clark had formed a FLOWTRAN Users Group for academics. This led, in 1977, to the publication of "Exercises in Process Simulation Using FLOWTRAN", which contained 27 simulation problems.

During the period 1974 to 1983, on the average, some 21 schools per year utilized the UCS Service bureau at a total cost of approximately $30,000/year. Some 59 schools used FLOWTRAN on UCS. before it was withdrawn on January 15, 1986 (Leesley and Pollicoff, 1978a,b; Rosen 1979; Hughes and Biegler, 1982)). In order to reduce the cost of using FLOWTRAN, schools requested that FLOWTRAN be made available on their own computers. In November, 1983 Monsanto announced that CACHE, after converting the code to run on a specified target machine, would distribute tapes to schools requesting FLOWTRAN. Converters in

Sales of *FLOWTRAN Simulation - An Introduction* by By Seader, Seider and Pauls (1975-1993)

Approximately 15,000

Tapes of FLOWTRAN Object Modules Distributed:

United States	141
Canada	11
Other Foreign	38
Total	190

Educational Institutions in Foreign Countries Receiving FLOWTRAN:

Argentina	Columbia	Korea	United Arab Emirates
Australia	Germany	Kuwait	Uruguay
Belgium	Greece	Malaysia	Yugoslavia
Brazil	India	Mexico	
Chile	Italy	Portugal	
China	Jordan	Spain	

Table 1. Data for the Monsanto/CACHE FLOWTRAN Project (Seader, 1994)

university centers were asked to carry out the conversion for their machine. Their version, in turn, would be the version distributed to others with the same machine.

The version of FLOWTRAN distributed by CACHE and used by students and instructors remained the same until 1987. By that time, versions of the program had been prepared for 14 different computer operating systems. The most popular versions were for DEC VAX VMS minicomputers and IBM CMS mainframes. In 1987, Monsanto provided three new process operation subroutines and L. T. Biegler provided an add-in optimization package capable of performing simultaneous optimization and convergence of control and recycle loops using Broyden's method. This package required a quadratic programming algorithm, QPSOL, from Stanford University, whose use was arranged for by G. V. Reklaitis. The third edition of the FLOWTRAN textbook, in 1987, included a chapter on optimization of process flowsheets by L. T. Biegler.

The impact that FLOWTRAN has had on university design education has been well documented (Clark and Sommerfeld, 1976; Edgar et al., 1985, 1988). Key to this impact, no doubt, has been the textbook *FLOWTRAN Simulation - An Introduction* (Seader et al., 1974) and the exercise book (Clark et al., 1980). Table 1, gives statistics on the distribution of FLOWTRAN literature and tapes in the Monsanto/CACHE project.

Summary

In 1993 (AIChE Annual meeting, St. Louis) the CACHE Corporation presented an award to Monsanto for its contribution of FLOWTRAN to chemical engineering education.

Monsanto plans to deposit many of the original historical documents in the Beckmann Institute for the History of Chemistry at the University of Pennsylvania.

Acknowledgement

The author wishes to thank Monsanto Co. for permission to publish this article. The comments and suggestions of J. D. Seader, A. H. Larsen, J. R. Deam and H. H. Chien are much appreciated.

References

AIChE Staff, Institute News: Monsanto Aids CACHE. *Chemical Engineering Progress* 70, 75 (1974).

Biegler, L. T. and R. R. Hughes, Approximate Programming of Chemical Processes with Q/LAP. *Chemical Engineering Progress* **77**, 41-57 (Apr 1981).

Cavett, R. H., Physical Data for Distillation Calculations-Vapor-Liquid Equilibria. 27th Midyear Meeting of API Division of Refining, San Francisco (May 1962).

Cavett, R. H., Application of Numerical Methods to Convergence of Simulated Processes Involving Recycle Loops. 28th Midyear Meeting of API Division of Refining, Preprint 04-63, Philadelphia (May 1963).

Cavett, R. H., Computer Physical Property Library, Part II. Enthalpy. Final Report 64-85, Pure Oil Company, Crystal Lake, Illinois (Feb. 1964a).

Cavett, R. H., Review and Critique of Relevant Mathematical and and Computer Methods for the Analysis of Recycle Processes. Report 64-183, Pure Oil Company, Crystal Lake, Illinois (Apr. 1964b), presented AIChE National Meeting, Pittsburgh (May 1964).

Cavett, R. H., Computer Physical Property Library, Part III. K-Values. Report 64-613, Pure Oil Co., Crystal Lakes, Illinois (Oct 1964c).

Cavett, R. H., Computer Physical Property Library, Part IV. Entropy and Volume. Report 64-633, Pure Oil Co., Crystal Lakes, Illinois (Oct 1964d). Cavett, R. H., FLOWTRAN Physical Properties. 49th NGPA Annual Convention, Denver (Mar 1970).

Cavett, R. H., Free Format Input for FLOWTRAN. Report, Computerized Engineering, Monsanto Co., St. Louis (Apr 1972a).

Cavett, R. H., Monsanto Physical Data System, Cryogenic Physical Properties. Report, Monsanto Enviro-Chem, St. Louis (Aug 1972b) presented at AIChE Annual Meeting, New York (Nov 1972b).

Cavett, R. H., A New Method of Solving Multiple Nonlinear Algebraic Equations. Ph.D. Thesis in Chemical Engineering, Arizona State University, (Aug 1993).

C&EN Staff, Process designers get more software. *Chemical & Engineering News,* (March 30, 1970).

Chemical Week Staff, To each engineer, his own computer. *Chemical Week* (March 18, 1970).

Chien, H. H. and C. B. Cobb, FLOWTRAN Applications. AIChE Symposium, St. Louis (March, 1969).

Chien, H. H., A Multiphase Algorithm for Single Variable Equation Solving. *Journal of the Institute Mathematics and Applications* **9**, 290 (1972a).

Chien, H. H. and H. R. Null, Generalized Multicomponent Equation for Activity Coefficient Calculation. *AIChE Journal* **18**, 1177 (1972b).

Chien, H. H., A Rigorous Calculation Method for Minimum Stages in Multicomponent Distillation. *Chemical Engineering Science* **28**, 1967 (1973).

Chien, H. H. and A. H. Larsen, Calculation of Multistage Multiphase Reacting Systems in Crystallization. *Industrial and Engineering Chemistry Process Design and Development* **13,** 299 (1974).

Chien, H. H., A Rigorous Method for Calculating Minimum Reflux Rates in Distillation. *AIChE Journal* **24**, 606 (1978).

Clark, J. P. and J. T. Sommerfeld, Use of FLOWTRAN Simulation in Education, *Chemical Engineering Education*, 90-92 Spring (1976).

Clark, F. P., T. P. Koehler and J. T. Sommerfeld Eds, *FLOWTRAN Exercises,* CACHE Corporation, Ann Arbor (1980).

Edgar, T. F., R. S. H. Mah and G. V. Reklaitis, Use of Computers in Chemical Engineering Education. *Chemical Engineering Progress*, **81**, 9-13 (1985).

Edgar, T. F., R. S. H. Mah, G. V. Reklaitis and D. M. Himmelblau, Computer Aids in Chemical Education. *CHEMTECH* **18**, 277-283 (1988). Harris, R. E., FLOWTRAN: An Approach to a Computer Aided Process Analysis and Design Tool, Talk to American Chemical Society, Los Angeles, (March 29, 1971).

Henley, E. J. and E. M. Rosen, *Material and Energy Balance Computations.* John Wiley, New York (1969).

Hughes, R. R. and L. T. Biegler, *CACHE Use of FLOWTRAN on UCS,* CACHE Corporation, Ann Arbor (1982).

Lang, V-D. and L. T. Biegler, A Unified Algorithm for Flowsheet Optimization. *Computers and Chemical Engineering* **11**, 143-159 (1987).

Larsen, A. H., Computerized Liquid Enthalpy Correlation for Process Simulation, Paper Presented at AIChE St. Louis Section Symposium '77 (Mar. 1977).

Larsen, A. H., FLOWTRAN for Process Simulation. Part IV, Chapter 22 in M. E. Leesley, Ed. Computer-Aided Process Plant Design, Gulf Publishing Co., Houston (1982).

Leesley, M. E. and H. S. Pollicoff, Evaluating Process Simulator Software. *Computer Aided Design* **10**, 155 (May 1978a).

Leesley, M. E. and H. S. Pollicoff, Comparing CONCEPT and FLOWTRAN Using a Process Evaluator. *Computer Aided Design* **10**, 289 (Sept 1978b).

Monsanto Enviro-Chem Systems Inc., FLOWTRAN (Circa 1972).

Motard, R. L., M. Shacham and E. M. Rosen, Steady State Chemical Process Simulation. *AIChE Journal* **21**, 417-435 (1975).

Parker, A. L., Chemical Process Optimization by Flowsheet Simulation and Quadratic Approximate Programming. Ph.D. Thesis in Chemical Engineering, University of Wisconsin (1979).

Parker, A. L. and Hughes, R. R., Approximation Programming of Chemical Processes, 1: Optimization of FLOWTRAN Models. *Computers and Chemical Engineering* **5**, 123-133 (1981a).

Parker, A. L. and Hughes, R. R., Approximation Programming of Chemical Processes, 2: Computational Difficulties. *Computers and Chemical Engineering* **5**, 135-141 (1981b).

Pauls, A. C., Private Communication (circa, 1993)

Rorschach, R. L., and R. E. Harris, Process Simulation Made by Computer. *Oil and Gas Journal* **68**, 33 (Aug 17, 1970).

Rosen, E.M., A Machine Computation Method for Performing Material Balances, *Chemical Engineering Progress* **58**, 69-73 (1962).

Rosen, E. M., response letter of June 17, 1964 to letter of R. H. Cavett dated May 27, 1964.

Rosen, E. M. and A. C. Pauls, Computer Aided Chemical Process Design: The FLOWTRAN System. *Computers and Chemical Engineering* **1**, 11-21 (1977).

Rosen, E. M., Letter to the Editor. *Computer Aided Design* **11**, 251 (Sept. 1979).

Rosen, E. M., Steady State Chemical Process Simulation: A State-of-the-Art Review in R. G. Squires and G. V. Reklaitis, eds. *Computer Applications to Chemical Engineering, Process Design and Simulation*, ACS Symposium Series 124, ACS, Wash, DC 3-36 (1980).

Rosen, E. M. and D. J. Kaufman, Computers, Design and Monsanto. *Computers in Industry* **2**, 41-57 (1981).

Sanderson, R. V. and H. H. Chien, Simultaneous Chemical and Phase Equilibrium Calculation. *Industrial and Engineering Chemistry Process Design and Development* **12**, 81 (1973).

Seader, J.D., Education and Training in Chemical Engineering Related to the Use of Computers. *Computers and Chemical Engineering* **13**, 377-384 (1989).

Seader, J. D., Private Communication (Feb. 17, 1994)

Seader, J. D., W. D. Seider and A. C. Pauls, *FLOWTRAN Simulation - An Introduction*. CACHE Corporation, Ann Arbor (1974, 1977, 1987).

Wegstein, J. H., Accelerating Convergence of Iterative Proceses. *Communications Assoc. Computing Machinery* **1**, 9 (1958).

AN EXPERT is a man who will know tomorrow why the things he predicted yesterday didn't happen today.

Installation Instructions; History/Content of 25th-Anniversary CACHE CD-ROM

By Peter R. Rony, Virginia Tech
Michael B. Cutlip, University of Connecticut

TO: Users of the CACHE Corporation 25th Anniversary CD-ROM

FROM: Peter R. Rony, Editor,
RONY@VTVM1.CC.VT.EDU

Michael B. Cutlip, Task Force Member and previous Chairman of CACHE,
MCUTLIP@UCONNVM.UCONN.EDU

INSTALLATION INSTRUCTIONS (December 23, 1994 version)

Please follow these instructions precisely if you are not experienced with Microsoft Windows.

Microsoft Windows

There is a Graphical User Interface (GUI) that runs under Windows to provide the easiest access to the CACHE CD-ROM contents. Users are advised to follow the instructions given below in order to execute the file, **25cache.exe**; your system maps your CACHE CD- ROM drive to logical drive **Z:** These instructions assume that you have previously installed your system CD-ROM drivers and that the system currently considers "g" (any letter) as your system CD-ROM drive. This setup procedure retains your original CD system drive "g" and adds drive **Z:** for the CACHE CD-ROM, thereby allowing the CACHE software to locate the needed files on drive **Z:** (Additional details of this mapping process are discussed on the CD-ROM as the first item under the *Readme* menu.) Be sure to place the CACHE CD-ROM into your CD ROM drive before proceeding.

(1) Add **lastdrive=z** as the last line of your *config.sys* file and **immediately reboot your computer**.

(2) While in DOS, establish the system CD-ROM drive as logical drive **Z:** by first creating a dummy subdirectory, *c:\dummy*, and then entering the following two DOS command lines: (note that the original system CD-ROM drive remains valid)

subst z: c:\dummy

assign z=g

[where "g" is your system-designated CD-ROM drive letter]

If you get the message "Bad command or file name" after you enter the assign command above, then this utility is not yet available in your DOS subdirectory. Please use the following DOS command to copy the assign file from the CACHE CD-ROM to your DOS file before repeating the assign command given above:

copy g:\dos_60\assign.com c:\dos

[where "g" is your system-designated CD-ROM drive letter]

Two subdirectories are available on your CACHE CD-ROM disc, **\DOS_60**, which contains the DOS 6.22 version of assign.com, and **\DOS_50;**, which contains the DOS 5.0 version of *assign.com*.

(3) XCOPY all of the files in g:*cachepif* to subdirectory *c:\windows\cachepif* by executing the DOS command line:

xcopy g:\cachepif c:\windows\cachepif

[where "g" is your system-designated CD-ROM drive letter]

(4) Enter Windows 3.1 (or Windows for Workgroups 3.11) and do one of the following although (c) is preferred for frequent use of the CACHE CD-ROM:

(a) Go to the File Manager for logical drive Z:\ and double-click the mouse button on file *25cache.exe* (in the root directory of **Z:**).

(b) Go to the File menu, click on the Run item, and then enter z:\25cache.exe

(3) XCOPY all of the files in *g:\cachepif* to subdirectory *c:\windows\cachepif* by executing the DOS command line:

xcopy g:\cachepif c:\windows\cachepif

[where "g" is your system-designated CD-ROM drive letter]

(4) Enter Windows 3.1 (or Windows for Workgroups 3.11) and do one of the following although (c) is preferred for frequent use of the CACHE CD-ROM:

(a) Go to the File Manager for logical drive Z:\ and double-click the mouse button on file *25cache.exe* (in the root directory of **Z:**).

(b) Go to the File menu, click on the Run item, and then enter z:\25cache.exe

(c) Go to the Main window (Program Manager) where you should use the New option under File to create a New Program Group with description "CACHE CD-ROM" and **no entry** in the Group File blank. Then

Either: Immediately use the New option under File **a second time** to create a New Program Item with the following Program Item Properties:

Description:	25cache
Command Line:	25cache.exe
Working Directory:	z:\
Shortcut Key:	None

Or: Drag the 25cache.exe file from **Z:** drive (using File Manager) to the CACHE CD-ROM window to install this file.

Then: Double click on 25cache icon to execute the program.

(5) Please use the Install option from the 25cache CACHE CD-ROM menu line to read about and install the Acrobat Reader (by Adobe Systems) for Windows 2.0 at an early stage in your work with the CD-ROM.

(6) Reuse of the CACHE CD-ROM after computer rebooting or new startup requires repeating step (2), and then executing *25cache.exe* by double clicking on the 25cache Icon in the CACHE CD-ROM Window or double clicking on cache.exe in the File Manager -[Z:\] Window. Frequent users should place the commands in (2) in the autoexec.bat file for convenience. Remember to always reboot the computer immediately after modifying the autoexec.bat file.

(7) The CACHE CD-ROM display is designed to run when Windows 3.1 is configured for VGA display by selecting Windows Setup under Program Manager.

NOTE: Drive **Z:** is hard-coded in the Authorware GUI program on the CACHE CD-ROM, *25cache.exe*, so it is necessary for you to execute the *assign.com* and *subst.exe* commands in the manner described above.

Macintosh

The Macintosh folders are clearly indicated and easy to use. They will automatically come up on your desktop from your CD-ROM drive.

Technical Details of the CACHE CD-ROM

The objectives of the CD-ROM were: (a) to demonstrate new computer technology (e.g., images, audio, video, animation, sequential presentations, authoring software, and multimedia presentations) useful in chemical engineering education; and (b) to provide a set of CACHE Corporation software "deliverables" to the target audience — chemical engineering students and faculty — for the CD-ROM.

The CD-ROM disc contains a total of 678.2 megabytes — identical (to four significant digits) to the number of acres in Lake Wobegon of Garrison Keillor fame. There are 4443 Windows/DOS files contained in 248 subdirectories. The Graphical User Interface (GUI) is 17.3 megabytes in size, and can be compressed using PKZIP.EXE to under 2.0 megabytes. Updates to the GUI, called 25CACHE.EXE, are planned; they will be distributed over the World Wide Web using a WWW server located in the chemical engineering department at Virginia Tech.

The technical keywords that describe the CACHE CD-ROM are **graphical user** interface (**GUI**), **perpetual menu**, and **launcher**. Macromedia, Inc. (San Francisco) Authorware for Windows version 2.01 was selected as the authoring software for the CD-ROM; Authorware organized information about and access to most of the files on the CD-ROM disc. Three reasons for selecting Authorware were: (1) the minimal need for scripting during the design of the GUI; (2) the ability to provide CD-ROM users with runtime versions of Authorware for Windows at no cost; and (3) the availability of an Authorware site license to the editor.

An Authorware **perpetual menu** was selected as the user navigation strategy—the organizing user interface—for the GUI. A perpetual menu consists of menu categories (located in a menu bar) that can be accessed at any point in the GUI. Such a strategy facilitates the

ability of a user to browse the available features of the CD- ROM. The ten perpetual menu categories included: *File, Readme, Install, Tutorial, Previews, CACHE, 3M, Depts, Software, and Internet.* Extensive nested-tree navigation structures were minimized.

The Authorware GUI serves primarily as a **launcher**, that is, a user selects a desired file and uses a mouse to press a key on the screen, thus launching: external application software, a software demo, a software working model, or an Acrobat Reader PDF file. Upon exit from the external software, a graceful return is made to the Authorware GUI.

Specific features of the CACHE CD-ROM include the following:

Examples of authoring software: 25CACHE.EXE in its entirety, 3M demo on optical media technology, Brigham Young demo, Media Vision demo, several Macromedia demos, Queen's University demo, Asymetrix demos, and University of Michigan multimedia presentations. Also, Macintosh hypertext presentations by Arizona State University and University of Michigan.

Examples of Adobe Acrobat technology: Install versions of Adobe Acrobat for Windows, Macintosh, and DOS version 2.0; Acrobat PDF versions of CACHE software manuals, Internet Windows client software manuals, Shakespeare's sonnets, UM Interactive Module manuals, and Lotus Freelance presentation on What is Chemical Engineering.

CACHE executable software: Polymath version 3.0.1, Picles versions 3.1 and 4.0 (beta), Digest versions 2.1 and 2.2 (beta), and Chemsep version 2.30.

Examples of Internet clients: Tutorials on the World Wide Web and how to create a WWW server; brief descriptions of Internet for Windows client software such as Mosaic, Hampson's Gopher, QWS3270, WS_FTP, Eudora, Trumpet Winsock, Trumpet, and others.

Examples of digital video: Install versions of Apple QuickTime for Windows and Macintosh, University of Michigan multimedia presentations, Wisconsin MPEG video, and examples of Microsoft Video.

Examples of digital audio: 3M demo on optical media, University of Michigan multimedia presentations

Examples of chemical engineering software: Aspen Technologies' MAX and AspenPlus, Simulation Sciences' Hextran, HTRI demo, Codeware demo, Molecular Design Ltd. ISIS demo, Keithley/Metrabye VDAS demo, Laboratory Technologies Vision demo, and Labtech Notebook demo.

Examples of multimedia software: Macromedia Authorware working model, Adobe Premiere, Asymetrix Multimedia Toolbook demo, Adobe Acrobat for Windows and Macintosh, and QuickTime for Windows and Macintosh.

CACHE Corporation Distributes 25th-Anniversary CD-ROM to Students and Faculty

San Francisco, November 13-18, 1994 — The CACHE Corporation (Computer Aids for Chemical Engineering Education) distributed at no cost, to students and faculty members, nine-hundred-fifty (950) 25th-Anniversary CACHE CD-ROM discs at the 1994 Annual meeting (San Francisco) of the American Institute of Chemical Engineers (AIChE), November 13-18, 1994.

625 discs were distributed to students at the AIChE student chapter breakfast held at the San Francisco Hilton and Towers Hotel on November 13; 325 additional discs were distributed to faculty and guests attending the CACHE 25th Anniversary Reception on November 16, 1994.

Production of the 25th-Anniversary CACHE CD-ROM was sponsored by 3M Pre-recorded Optical Media (Menomenie, Wisconsin)—who provided artwork, mastering services, and 1200 free discs—and financially co-sponsored by many of the contributors to the CD-ROM, including Aspen Technologies, Simulation Sciences, HTRI, Molecular Design Ltd.. Adobe Systems, Asymetrix, Keithley/Metrabyte, Media Vision, Hypercube, Douglas-Stewart, Codeware, AIChE, Laboratory Technologies, Brigham Young, Queen's University, Arizona State, Colorado State, Louisiana State, Purdue, Michigan, Wisconsin, and Virginia Tech. Macromedia, Inc. provided a runtime version of Authorware for Windows 2.0 for each disc. **CACHE gratefully acknowledges the sponsorship by each and every one of these organizations.**

Ordering CD-ROMs

To order CD-ROMs, complete the order form in the back of this newsletter and send with payment to CACHE Corporation.

CD-ROMs are presently being sold in bulk to AIChE Student Chapters at a price of $5 per copy. AIChE Student Chapters may, in turn, sell these CD-ROMs for an amount up to $10/copy in order to generate funds for student activities.

Individual copies are $10/copy for students and professors and $35/copy for non-academics.

The Chemical Reactor Design Tool

By Bruce A. Finlayson, University of Washington

The Chemical Reactor Design Tool is a set of computer programs that permits students to design chemical reactors containing realistic transport effects that are frequently present. The interface is written in X-windows so that the student can include complications easily; the program automatically uses the correct, robust tools to solve the problem. Results are displayed graphically, making comparison studies especially easy. These programs were developed under the co-sponsorship of the National Science Foundation and the University of Washington. They have been made available to universities through CACHE Corporation.

CRDT Educational Goals

Introductory textbooks concentrate on problems that can be solved analytically. Recent textbooks include material for problems that can be solved with an ordinary differential integrator such as batch reactors and simple, plug flow reactors; extensions from one reaction to several reactions are possible, but are also time consuming. When attempting to solve real problems, students are faced with several difficulties, which tend to be in manipulation and book-keeping rather than conceptualization. Phenomena that might be important include:

- multiple reactions (lots of bookkeeping)
- temperature of catalyst and fluid may be different
- there may be internal mass transfer (requires solving an effectiveness factor problem)
- there may be cooling at the wall (leads to radial dispersion)

Students and design engineers may not be able to make realistic estimates of which phenomena must be included. In some cases it is necessary to calculate with the suspected phenomena included to determine whether it is important. This process is often difficult as each phenomena creates problems which require special techniques to solve. CRDT uses: ordinary differential equation integrators, the orthogonal collocation method, the finite difference method, techniques to convert partial differential equations to sets of ordinary differential equations, iterative techniques to solve large sets of nonlinear equations, and linear programming methods to guide initial guesses for iterative techniques. These methods are all transparent to the user.

Key Aspects

Users can examine effects very easily and make their own deductions about the importance of physical phenomena. Phenomena that can be included easily are:

- different reactors: CSTR, batch, plug flow
- axial dispersion, radial dispersion
- Intraparticle heat and mass transfer
- Significant mole changes
- Significant pressure changes

Cases where it makes a difference are:

- Selectivity, especially in non-isothermal cases
- Non-isothermal problems

Neither of these two cases is too difficult for the student to do, but there is generally not enough time to do so. However, some of these complications are necessary in design problems.

Textbook Supplements

Important features include Textbook Supplements, which provide any new reactor design equations using the notation of standard textbooks. Textbook supplements (90 pages long) have been created for Fogler [1] and Hill [2], two widely used reaction engineering textbooks. Because the programs are much more general than can be treated in these books, it is necessary to explain the problems employing the notation of the textbook used by the student. These textbook supplements: give a program description, generate the equations in the notation of the textbook, work problems from the textbook, work extensions of problems in the textbook, give a list of possible extension problems (for the instructor to assign), and provide a summary of the equations. In addition, a four-page handout provides: hints on the best way to approach problems, summaries of the equations for easy reference, and standard correlations for some of the transport properties. This handout is designed for quick reference while using the program.

Table of Contents from the Student's Manual

1. Introduction
2. Develop Equations in the Notation of the Textbook
3. Description of the Computer Code
4. Textbook Examples Solved using CRDT
5. Problem Extensions
6. Extension Problems Solved with CRDT
7. Literature Problems
8. User's Manual
   - Notation
   - Reaction Rates
   - Heterogeneous Reactions
   - How to Find Species Concentrations from Molar Flow Rates
   - Velocity Calculations
   - Pressure Drop Calculations
   - Effectiveness Factor
   - Equilibrium Reactions
   - Dimensional Form of Equations
   - Non-dimensional Form of Equations
   - Methods of Solution
   - Plotting - Line Plots
   - Plotting - Contour Plots or 3D Views

Table of Contents from the Instructor's Manual

1. Introduction
2. Input Instructions
3. Output Format and Plotting Instructions
4. REACSET Screen Summaries
5. Newton-Raphson and Homotopy Methods
6. Relate Different Formulations for Reactors with Variable Volume or Number of Moles
7. Relate Different Formulations for Non-isothermal Chemical Reactors
8. Derivation of Non-dimensional Form of Equations
9. Installation

Design Decisions

Design decisions may revolve around conflicting constraints, none of which can be easily handled if one writes the program themselves.

- Use a small catalyst diameter to avoid diffusion resistance, but this increases the pressure drop. A recycle compressor may be an expensive component in a gas-phase reaction system.
- An adiabatic reactor avoids radial dispersion, but the temperature rise may be too big; cooling at the wall usually makes radial dispersion important. By using the Chemical Reactor Design Tool these realistic complications can be treated by the student-designer.

Output

Output is presented graphically in addition to the written output. The user can call upon the following plots.

- line plots
- 3D perspective views
- 2D contour plots
- solution variables: concentration, molar flow rate, temperature, wall flux, solid temperature, difference between fluid and solid temperatures, solid concentration, difference between fluid and solid concentrations
- each term in the equations: diffusion, reaction, convection terms These plots are created automatically, but the user has some control over them either before they are created (contour and 3D views) or after they are created (line plots).

Example Equations

Typical equations are listed here to remind the reader of the type of equations. The actual Chemical Reactor Design Tool allows up to 20 components and temperature to be variables. The user need only supply a reaction rate subroutine, written in FORTRAN, with a model subroutine made available. The program allows significant mole changes and pressure drops at the click of a button.

Batch Reactor

$$\frac{dN_j}{dt} = \rho_B RA(j) V_R; \text{ initial conditions: } N_j(0) = N_{j0}$$

CSTR

$$F_{j0} - F_j + \rho_B RA(j) V_R = 0$$

Plug Flow Reactor

$$\frac{dF_j}{dV_R} = \rho_B RA(j); \text{ inlet conditions: } F_j(0) = F_{j0}$$

Axial Dispersion

$$\varepsilon D_{ezj} \frac{d^2 C_j}{dz^2} - \frac{d(uC_j)}{dz} + \rho_B RA(j) = 0$$

$$\text{Boundary conditions: } \frac{d(C_j)}{dz}(z = L) = 0$$

$$-\varepsilon D_{ezj} \frac{dC_j}{dz}(z = 0) = u_{in} C_{j,in} - u(z - 0) C_j(z = 0)$$

Radial Dispersion

$$\frac{1}{A_c}\frac{\partial F_j}{\partial z} = \frac{\varepsilon D_{erj}}{R^2}\left(\frac{\partial^2 C_j}{\partial r'^2} + \frac{1}{r'}\frac{\partial C_j}{\partial r'}\right) + \rho_B RA(j)$$

Boundary conditions: $\frac{\partial C_j}{\partial r'} = 0$ at $r' = 0$

$$\frac{-\varepsilon D_{erj}}{R}\frac{\partial C_j}{\partial r'} = k_w[C_j - C_{wj}] \text{ at } r' = 1$$

Initial conditions: $F_j(r', 0) = F_{j0}(r')$

Example. In Chapter 9 of Fogler the following reaction system is considered to study selectivity.

$$R \rightarrow D \qquad r_1 = k_1 C_R^{\alpha_1}$$

$$R \rightarrow U \qquad r_2 = k_2 C_R^{\alpha_2}$$

R is the reactant, D is the desireable product, and U is the undersirable product. Fogler argues that if $\alpha 1 > \alpha 2$ a plug flow reactor is preferred, whereas if $\alpha 1 < \alpha 2$ a CSTR is preferred. The CRDT can be used with the rate expressions:

```
k1 = 1.    k2 = 1.    alpha1 = 2.    alpha2 = 1.
r1 = k1*c(1)**alpha1
r2 = k2*c(1)**alpha2
ra(1) = -r1 -r2
ra(2) = r1
ra(3) = r2
```

With F = 1 mol/s, V = 2.2 l, we get the following output concentrations:

CSTR	R = 0.2644	D = 0.1538	U = 0.5817
PFR	R = 0.05865	D = 0.3052	U = 0.6362

The PFR is preferred slightly, as argued. Now suppose the reaction rate term is

$$r_i = \frac{k_i C_R^{\alpha_i}}{1 + K_1 C_R + K_2 C_D + K_3 C_U}$$

The argument about the preferred reactor is now much harder to make, but the calculations can answer the same question. For K1 = 0, K2 = 0, K3 = 2 we merely need to inset the statements:

```
den = 1 + 2.*c(3)
r1 = r1/dem
r2 = r2/dem
```

and redo the calculations. The results are

CSTR	R = 0.3833	D = 0.1709	U = 0.4458
PFR	R = 0.1595	D = 0.2953	U = 0.5451

The PFR is still preferred, but the ratio of D in the two reactors is closer. If there is mass transfer resistance, also, we merely need to click a button in CRDT, set the coefficients (2.0 was used) and obtain

CSTR	R = .4704	D = 0.1373	U = 0.3923
PFR	R = 0.2947	D = 0.2147	U = 0.4905

Now the ratio of concentrations of D out of the two reactors is even closer. Likewise if axial dispersion can be included in a PFR by clicking a button and inserting the value of e Dez (0.1 was used).

PFR-axial	R = 0.5566	D = 0.1480	U = 0.2954

The effect of axial dispersion to to make the PFR more like a CSTR. Comparisons of these models are shown in Figures 1-3, which are obtained from CRDT by the click of a button. Once the rate routine was created and checked it took less time to obtain all these results than it took you to read about them.

Suppose the reactor has a permeable membrance at the wall and the membrance permits the removal of the undesirable component (which inhibits the reaction rate). Then we use a two-dimensional model (click a button), set the radial dispersion coefficients (0.1 was used) and wall coefficient for component 3 (1 was used). The

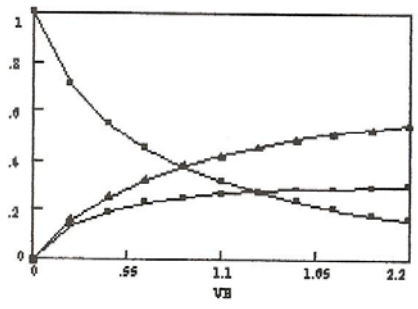


Figure 1. Concentrations for Plug Flow Reactor (curves can be identified from the concentration out and the text)

profile of the concentration of the desirable component is shown in Figure 4, while contours of the concentration of the undesireable component is shown in Figure 5. As expected, it is removed at the wall and this increases the reaction rate.

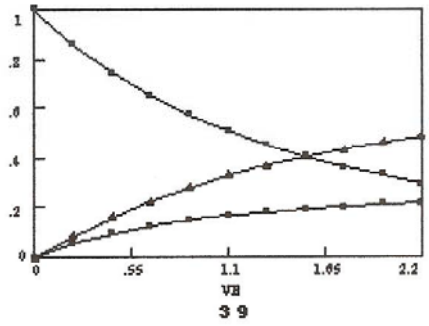


Figure 2. Concentrations for Heterogeneous Plug Flow Reactor

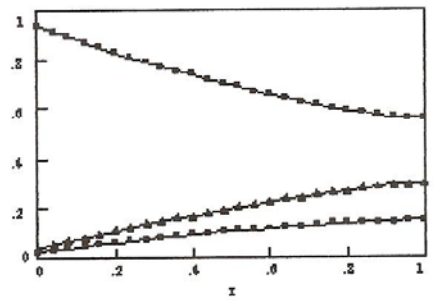


Figure 3. Concentrations for Plug Flow Reactor with Axial Dispersion

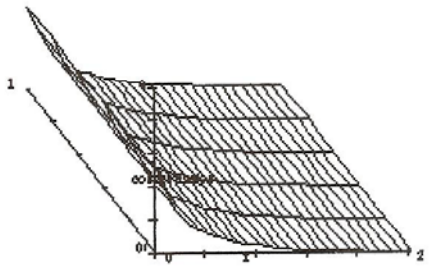


Figure 4. Concentration of Reactant in 2D Reactor

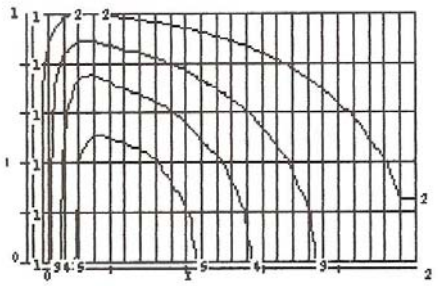


Figure 5. Concentration of Undesireable Component in 2D Reactor

Installation

The Chemical Reactor Design Tool is made available through CACHE. Orders for the program, Student's Manual and Instructor's Manual are placed with the CACHE office. The program is installed on your computer remotely by Professor Finlayson after notification by the CACHE office. The computer must run the Unix operating system. Currently modules are available for Digital, IBM, and Sun Computers. Modules for Hewlett-Packard computers will be made as time is available.

For More Information

Bruce A. Finlayson
Department of Chemical Engineering, BF-10
University of Washington
Seattle, Washington 98195
206-543-2253; FAX 206-543-3778
finlayson@cheme.washington.edu

References

1. Fogler, H. S., Elements of Chemical Reaction Engineering, Prentice-Hall, 2nd ed. (1992).
2. Hill, C. G., Jr., An Introduction to Chemical Engineering Kinetics & Reactor Design, Wiley (1977).

Purdue-Industry Laboratory Modules

By S. Jayakumar, Purdue University

A series of computer simulations of state of the art industrial processes are developed to be used as part of the Chemical Engineering laboratory course. The emphasis of these modules is on planning of experiments and analysis of experimental data rather than detailed process design. Each module is posed as an industrial problem caused by a change of conditions in an existing process, requiring an "experimental" study to re-evaluate the characteristic constants of the process such as reaction rate constants, equilibrium constants, heat transfer and mass transfer coefficients, and phase equilibrium constants. The student teams are expected to design "experiments" (and run them on the computer) which will enable them to evaluate the needed constants.

After the constants are determined, the students can validate these by using them in a "computer simulation", and comparing the simulated and experimental results. After verifying that their constants are reliable, the students use these parameters to study an existing or new process focusing on aspects such as efficient startup, effect of scale up on heat transfer requirements, controllability at desired steady states, balancing catalyst decay with product yield, maximizing productivity of a range of products, or analyzing the performance in other specific ways.

The project begins with a twenty-minute video-taped "tour" of the process. Each project is given some degree of realism also by assigning financial budget and time constraints. The problems are open-ended in that the conditions of experiments, such as temperature, pressure, flow rates and compositions are under the students' control. The cost and the associated time duration of running experiments vary with the type of experiment and are also functions of operating conditions. Instructor-controlled statistical fluctuations are built into the simulations so that the results of duplicate experiments are not identical. The students must plan their experiments to obtain data from which, with proper analysis, the required constants may be determined without exceeding their budgetary and time constraints.

The following modules are currently available:

1. Amoco Resid Hydrodesulfurization Process
2. Eastman Chemical Methyl Acetate Reactive Distillation Process
3. Mobil Research and Development Corporation Catalytic Reforming Process
4. Dow Chemical Company Styrene-Butadiene Polymerization Process
5. Air Products Cryogenic Hydrogen Reactive Cooling Process

An additional module nearing completion is the Albemarle Ethylene Oligomerization processes. Work on a coffee decaffeination process simulation sponsored by Procter & Gamble is now in progress. This is presented in a multimedia environment integrating audio, video and animations to the process simulation, in the Macintosh platform. Current usage of these modules is 25 schools in 6 countries. Upon completion of an NSF sponsored project to port these modules from Sun to HP, DEC, IBM RISC and Silcon Graphics workstations, usage is expected to more than double.

Apart from the undergraduate design laboratory, the computer modules have also been used successfully in reactor design and process control courses in several schools. The programs are configured to run on Sun Sparc workstations with 12 MB of memory. Each module uses less than 10 MB of disk space. Features include easy to use point-and-click user interface and context-specific on line help. Interested readers are referred to the following articles for more information:

1. R. G. Squires, G. V. Reklaitis, N. C. Yeh, J. F. Mosby, I. A. Karimi and P. K. Andersen, Purdue-Industry Computer Simulation Modules - The Amoco Resid Hydrotreater Process. Chem. Engr. Educ. 25(2), 98-101, 1991.

2. R. G. Squires, P. K. Andersen, G. V. Reklaitis, S. Jayakumar and D. S. Carmichael, Multi-media Based Educational Applications of Computer Simulations of Chemical Engineering Processes. Comp. Appns. Engr. Educ. 1(1), 25-32, 1992.

3. S. Jayakumar, R. G. Squires, G. V. Reklaitis, P. K. Andersen and L. R. Partin, Purdue-Industry Computer Simulation Modules: (II) The Eastman Chemical Reactive Distillation Process. Chem. Engr. Educ., 27(2), 136-139, 1993.

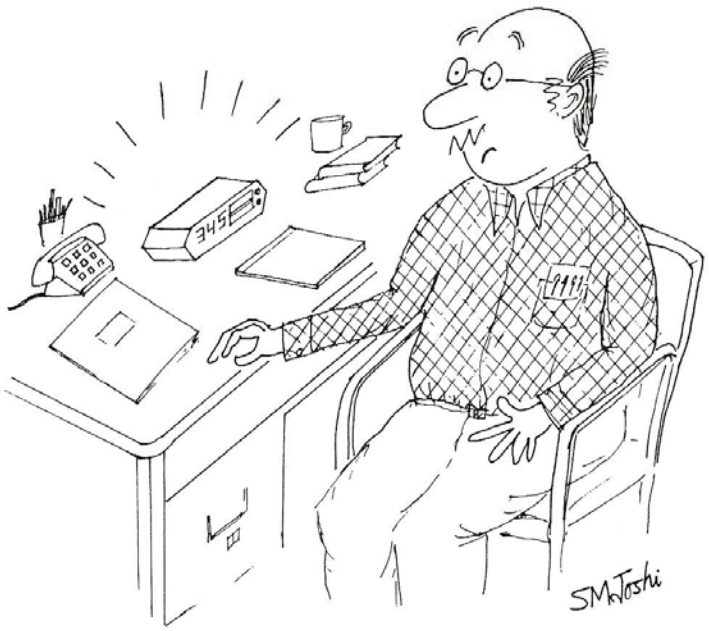
4. S. Jayakumar, R. G. Squires, G. V. Reklaitis, P. K. Andersen, B. C. Choi and K. R. Graziani, "The Use of Computer Simulations in Engineering Capstone Courses: A Chemical Engineering Example - The Mobil Catalytic Reforming Process Simulation," Int. J. Engr. Educ., 9(3), 243-250, 1994.

5. S. Jayakumar, R. G. Squires, G. V. Reklaitis, and K. S. Grassi, "Simulating The Air Products Cryogenic Hydrogen Reactive Cooling Process," Chem. Engr. Educ., Winter 1995 (In Press).

For More Information

To obtain the Purdue modules, please contact CACHE Corporation. For other information, please contact:

Dr. S. Jayakumar
School of Chemical Engineering
Purdue University
W. Lafayette, IN 47907-1283
E-mail: jai@ecn.purdue.edu.



"This is a special bulletin from KIDU, your news and weather station... some eigenvalues have reportedly broken out of the unit circle and are headed your way!"

IEEE Control Systems Magazine

Experiences with the Creation of a World Wide Web (WWW) Server

by Robert Bour, Virginia Tech (rbous@vt.edu)

NOTE:

Robert Bour, a junior chemical engineering student at Virginia Tech, wrote a series of four memoranda concerning the creation of a World Wide Web (WWW) server using Internet clients for Windows. The hardware was an IBM ValuePoint 486DX personal computer, substantially similar to the type of computer purchased by all freshmen and sophomore students in the College of Engineering (but without a double-speed CD-ROM drive or a Soundblaster 16 board).

To start the project, Rob first contacted a key Virginia Tech Internet guru, Dr. Harry M. Kriz, in the University Libraries. Harry has provided important contributions (in the Internet menu) to the 25th-Anniversary CACHE CD-ROM, and has recently contributed an excellent article, "Windows and TCP/IP for Internet Access," to the Summer 1994 issue of CAST Communications. Harry's second article, "Teaching and Publishing in the World Wide Web," was commissioned for the 25th-Anniversary CACHE CD-ROM. When subsequently placed on the Internet, there were 2500 inquiries during the first two weeks. All of Harry's articles are available on Internet and are updated periodically. For further details, contact him at HMKRIZ@VT.EDU.

The CACHE CD-ROM server should be available for testing at the end of March, 1995. The URL address will be http://rony.che.vt.edu (the server will not be operable when the computer is being used for a controls laboratory experiment and at several other times).

. . . . *Peter Rony (Professor of Chemical Engineering, Virginia Tech)*

MEMORANDUM NO. 1: THE PROPOSAL

TO: Peter R. Rony
FROM: Robert Bour

I propose the creation of an electronic forum for the exchange of ChE-related files, and perhaps ideas, at Virginia Tech. I envision two possibilities for such a forum:

WWW Site: As I understand it, you have access to the software and hardware required to create such a site. Since distributed software must be limited to enrolled Tech students, the site would have to require a user ID and password. Such a site would be an excellent way to distribute software since all students have the WNET suite required to login. I have some experience in using the Internet but, because of my extensive background in telecommunications, I should be able to learn all I need to know quickly. I assume that the software packages you refer to have limited documentation. I would be happy to take over this project or to help you with its startup.

Local BBS: A second option is a local bulletin-board system. This would solve the problem of unauthorized students logging on, since only Tech ChEs would be calling. File transfers would occur with the same ease and speed as a WWW site. The primary advantage of this possibility is the 'message base' feature that would allow students a true electronic forum to gather and exchange ideas. It would also facilitate communications between faculty and students, as many students are unavailable during professor s office hours. The biggest disadvantage of this idea is that the only terminal emulation program that students have is Terminal in Windows. We would have to distribute a better shareware term program, which is one extra step that the WWW Site does not require. I have access to BBS software, and I would be happy to run the bulletin board off of my CPU.

MEMORANDUM NO. 2: SETTING UP A WWW SERVER

TO: Peter R. Rony
FROM: Robert Bour

I write this memo to document my work in setting up the Virginia Tech chemical engineering World Wide Web (WWW) server, and to personally testify to Harry Kriz's claims [1] that setting up such a server is a very easy thing to do. I will provide specific information in this memo. For a general overview of the WWW, I recommend the Harry Kriz's first article [2].

I used three pieces of software in the setup of the WWW server. They were Trumpet Winsock, NCSA's HTTPd for Windows version 1 .3, and an HTML editor called HoTMetal. All of these are available via

anonymous FTP. Virginia Tech has a site license for Trumpet Winsock.

Trumpet Winsock was already set up on the computer I worked with, so I have no experience with it. My understanding is that it only involves the input of a few addresses and is fairly easy to initialize. The addresses that I used for this initialization process are:

My IP Address NetMask: 255.255.252.0 Gateway: 128.173.164.1 Name Servers: 128.173.4.247, 128.173.4.113, and 128.143.2.7

HTTPd was extremely easy to install. The entire process took me under an hour. I created a directory for it, unzipped the archive with a -d extension, added one environment variable to the AUTOEXEC.BAT file, and it was ready to go. The package comes with default home pages so that you can test the server before having to create a lot of HyperText documents. The change to the AUTOEXEC.BAT file was the addition of the following line:

set tz= est4edt

HoTMetal was even easier to install, but a bit more challenging to use since it requires a basic knowledge of the HyperText Markup Language (HTML). Even this is not very difficult. Several tutorials and primers are available via anonymous FTP.

In conclusion, setting up a WWW server is indeed a very simple process, with potential to yield great rewards. I suggest that if you have an interest in setting one up, pursue it without fear.

References

[1] Harry M. Kriz, "Teaching and Publishing in the World Wide Web," Virginia Tech University Libraries, October 1994. 2. Harry M. Kriz, Windows and TCP/IP for Internet Access, Virginia Tech University Libraries, current 1994 version.

MEMORANDUM NO. 3: ADDING HTML FILES TO A WWW SERVER

TO: Peter R. Rony
FROM: Robert Bour

I write this paper to document the completion of the setup process for the Virginia Tech chemical engineering World Wide Web (WWW) Server. At the end of the previous memo, the server was up and running with the pre-installed HyperText Markup Language (HTML) documents. The final stage of the setup was to install ChE-related HTML pages for viewing by our audience.

The pages I selected were already created and running on another server on campus. I downloaded the pages from that server and installed them on ours (this procedure may be useful to you if you ever come across a document that you would like to partially duplicate on your own server). The steps were as follows:

1. Using Mosaic, I connected to the server containing the desired HTML documents. 2. Under the Options menu, I chose "Load to Disk." 3. From the server's screen, I selected the highlighted document I wished to retrieve. 4. Mosaic prompted me for a filename, which I provided, and the document was downloaded to my hard drive.

It is useful to note at this time that if you want to download the current page on the Mosaic screen, you should choose Load to Disk as above; then, from the Mosaic menu, choose Navigate, Reload.

Once all the documents were downloaded, the task of implementing them was trivial. I simply copied them to the document directory (in my case it was c:\httpd\htdocs) and replaced the old INDEX.HTM with the .HTM file associated with the "home page" of the copied server. Thus, when a call would come in to our server, the software (HTTPD) would load up the new INDEX.HTM file which would in turn load the other documents I downloaded as they were selected by the user. This was the most recent step in the setup of the ChE WWW Server.

Since it is not likely that you will find suitable HTML documents elsewhere on the Net, you will probably have to create your own. The procedures for this are available over the Internet. Harry Kriz recommends several HTML tutorials in his second WWW article [1].

The ChE WWW server is currently accessible through any WWW client such as Mosaic or Cello. The current procedure is to select File, Open URL, then type on the URL blank line, http://rony.che.vt.edu . This will connect you to our server, which is located in the undergraduate process controlsl aboratory in Hancock Hall. To those of you who endeavor to create your own servers, I wish you good luck, and I encourage you to contact me for help if you need it. My e-mail address is rbour@vt.edu.

References

[1] Harry M. Kriz, "Teaching and Publishing in the World Wide Web," Virginia Tech University Libraries, October 1994.

MEMORANDUM NO. 4: FILE TRANSFER TESTS OF THE WWW SERVER

TO: Peter R. Rony
FROM: Robert Bour

Now that the Virginia Tech chemical engineering World Wide Web (WWW) server has been running successfully for three months, I write to discuss further

applications of this site. Consistent with your goal to provide software updates to owners of the CACHE CD-ROM, I have successfully implemented and tested the file transfer capability of the server. We can now provide users with access to specific files in a simple and user-friendly environment. These are the steps a typical user would follow to obtain a file from the server.

1. From a WWW client such as Mosaic or Cello, the user connects by entering the URL address of the ChE WWW the server (currently http://rony.che.vt.edu).
2. Following hypertext cues, the user navigates to the page that contains the desired file (such as a software update).
3. From the Mosaic "Options" menu, the user selects "Load to Disk" (other WWW hypertext browser packages will have slightly different menu systems)
4. Following on-screen instructions, the user clicks on the highlighted filename and the file transfer begins.

With this capability, the CACHE Corporation can make updates to the 25th Anniversary CACHE CD-ROM available to all Internet users as they are completed. Please feel free to contact me for any additional information. My e-mail address is rbour@vt.edu. Thank you.

Evidence of Internet in the World Wide Web Server

by Harry M. Kriz (Harry_M_Kriz@vt.edu), University Libraries, Virginia Tech

NOTE:

I asked Harry Kriz, a major contributor to the 25th-Anniversary CACHE CD-ROM (see Internet menu), to comment on the statistics of his two WWW articles. Such statistics demonstrate the enormous publication potential and power of the Internet if you provide a valuable service (e.g., excellent articles of broad interest).

... Peter R. Rony, CACHE Electronic Communications Task Force

The Web server article [which was commissioned for the 25th-Anniversary CACHE CD-ROM] was accessed 2,300 times in the 6 weeks following my announcement of its availability in mid-October 1994. Word about it has spread, apparently, as interest in it continues. In January 1995 it was accessed nearly 600 times by individuals in about 20 countries. Also, some 3500 machines at commerical organizations and 2885 machines at U.S. educational institutions accessed the server.

In January 1995, my Web server distributed over 14,000 files totaling more than 80MB. Virtually all the traffic is generated by individuals who desire to read one or both of my two articles. The first is my "Web server" article and the second is my article on "Windows and TCP/IP." The latter was accessed 3766 times in January. The total number of files distributed is greater than the sum of readers for each article because each article consists of several files. The counts I report for each article are the total accesses of the Abstract page in each article, i.e., the entry point for the article.

My statistics generator just counts the total number of unique IP addresses or hostnames. At least some of the addresses are gateway or proxy machines that channel multiple users onto the Internet from behind organizational firewalls. All the server knows is the IP address (and hostname) of the machine that makes the contact. Addresses such as ccvax.fullerton.edu probably involve multiple users. Other addresses obviously represent a single individual.

I continue to be amazed at the reach I can achieve with this technology without any help from publishers or learned societies. Information about my article has spread around the world by word of mouth.

ANNOUNCEMENTS

1995 Conference Announcement

ISPE'95

Intelligent Systems in Process Engineering

July 9 - 14, 1995

Snowmass Village, Colorado
USA

CO-CHAIRS

George Stephanopoulos ..... MIT
James F. Davis . ..... Ohio State University
Venkat Venkatasubramanian ..... Purdue University

SCOPE

Over the past ten years we have witnessed a broadly-based and intense effort to bring ideas and methodologies from artificial intelligence into the scope of process systems engineering problems. Academic research and industrial practice have generated an impressive amount of work, which spans virtually every aspect of process engineering work, e.g., product and process development and design, process operations monitoring-diagnosis, process control, operations planning and scheduling, operator training, process hazards analysis and risk assessment, etc.

Ten years later, it is important to pose for a week and critically examine what happened during these ten years, especially in view of the fact that work in the area of intelligent systems has matured significantly and is becoming well-integrated into the main stream activities of process engineering.

In response to these challenges, CACHE Corp. is organizing ISPE '95 with the following objectives in mind:

(1) Bring together a diverse group of people with interests in modeling and simulation, design, operations/control, process management, computer science and technology, operations research, statistics, systems and control theory, all of which are components in the solution of process engineering problems.

(2) Present the state-of-the-art developments in intelligent systems for various areas of process engineering, and discuss the experience gained from industrial applications.

(3) Provide a forum for in-depth discussions between researchers and practitioners on the theoretical and practical challenges in developing and deploying intelligent systems in industry.

1996 Conference Announcement

CPC-V

Chemical Process Control - V

January 7-12, 1996

Granlibakken Conference Center
Tahoe City, California
USA

Sponsored by:
CAST Division of AIChE
and CACHE Corporation

STEERING COMMITTEE

Co-Chairs

Carlos Garcia ..... Shell Oil (USA)
Jeffrey Kantor ..... University of Notre Dame (USA)

Committee Members

Yaman Arkun ..... Georgia Tech (USA)
James Downs ..... Tenn. Eastman (USA)
Frank Doyle ..... Purdue University (USA)

Tom Edgar..... University of Texas (USA)
Manfred Morari ..... Caltech (USA)
John Perkins..... Imperial College (UK)
James Rawlings ..... University of Texas (USA)
Sigurd Skogestad..... University of Trondheim (N)

ABOUT THE CONFERENCE

Chemical Process Control - V (CPC-V) will bring together engineers and scientists from universities, the processing industries, government laboratories to assess and critique the current and future directions of research in chemical process control. CPC-V is the fifth in a series of international conferences held every five years commencing in 1976.

The goals of CPC-V are to:

(1) promote a vital, interactive discussion among a diverse group of experts regarding the state of the technology for chemical process control,

(2) assess current research,

(3) identify research opportunities for academics, government, and industry, and

(4) promote productive research collaborations.

Conference participants will have ample opportunity to interact with peers and experts in their fields.

CONFERENCE FORMAT

The conference will be organized into morning and evening technical sessions that address current issues in process control. The sessions will consist of two or three papers followed by a formal critique or discussion. There will be at least an hour in each session reserved for audience participation in the discussion. The afternoons will be free for further informal discussion. A session for contributed poster papers is also planned.

To achieve the conference goal of interactivity, there will be many opportunities discussion and social interaction. Meals will be taken together in the morning and evening. Breakout rooms will be available for afternoon discussion. The will be daily refreshments, an opening reception, and a Thursday night conference banquet.

CONFERENCE THEMES

The main themes of the conference include:

Surveys and Assessments of current research areas

- Robust & predictive control
- Modeling and identification
- Nonlinear & Adaptive control

New research directions

- Process monitoring and fault detection
- Sequence control, PLC's, and discrete event systems
- Sensors and data fusion

Surveys of control practice in industry

- Food processing
- Refining and chemicals
- Pulp & paper
- The nontraditional process industries

Status of Industry/Academic Research

Twenty years have passed since CPC-I. What has been accomplished, and what are the prospects?

- What have been the directions and results of government and industry sponsored research.
- When does joint academic/industry research pay off?
- What are the prospects?

LOCATION

The Granlibakken Conference Center is near Tahoe City, California, located on the north side of Lake Tahoe in high Sierra Nevada mountains. It is a one hour drive from the Reno Cannon International Airport in Reno, Nevada. The Reno airport is served by America West, American, Continental, Delta, Reno Air, Southwest, Sky West, United, and U.S. Air. There is a van service between the Reno airport and the Conference Center.

Winter recreational activities are available at the conference site and at nearby major ski resorts. The conference center offers a small ski facility with surface lift, ski rental shop, ski school, and cross-country ski trails. A free shuttle takes you to the Squaw Valley and Alpine Meadows ski areas.

Restaurants and fine shops are nearby. There is bus and boat service to the major tourist and entertainment areas located in South Lake Tahoe.

CONFERENCE SCHEDULE

The conference features nine technical sessions held in the mornings and evenings. Except on Thursday, afternoons will be free for informal discussions, demonstrations, or recreation. A Thursday afternoon poster session will be followed by an evening banquet.

The conference will begin with an evening session following dinner on Sunday, January 7, and conclude following lunch on Friday, January 12.

PARTICIPANTS / CONFERENCE SIZE

The conference will be limited to 150 participants. Prior experience has shown this to be the largest size that can reach the conference goals of cohesiveness and interactivity. The selection of participants will be based on applications; the criteria include diversity by employment (industry, academics, government), residence (US, Europe, Pacific Rim, emerging economies), and experience (graduate students, practitioners, and world class experts).

CONTRIBUTED PAPERS

A poster session is planned to accommodate contributed papers. Submission details will be provided to invited participants.

ACCOMMODATIONS AND MEALS

The Conference Center includes reasonably priced lodging in shared one, two, and three bedroom townhouses with kitchens. All units have TV and direct dial telephones; all bedrooms have private baths. Accommodations will be on the American plan with three full meals.

A variety of lodging packages will be offered, with a basic package to include all meals and double occupancy sleeping accommodations for $650. There will be moderate additional charges for single rooms, or for an accompanying spouse or guest sharing a room. Participants may wish to extend their stay and enjoy the scenic location and winter recreation.

Further details on room and board packages will be provided later to invited participants.

APPLICATION

Attendance at CPC-V will be limited and by invitation following receipt of this application. The conference fee ($675) will include registration, proceedings, opening reception, conference banquet, and daily refreshments.

Please complete and return this application, no later than July 15, 1995, to:

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

You will be notified by August 15, 1995 regarding an invitation.

Name: _____

Title: _____

Affiliation: _____

Mailing Address: _____

_____

_____

_____

_____

Telephone: _____

E-mail: _____

Fax: _____

Research Interests:

_____

_____

Main Interests in this Conference:

_____

_____

Heat Transfer Research, Inc. (HTRI) Releases New Educational Tool

By R. S. Kistler, Heat Transfer Research, Inc.

HTRI has released a special version of their ST Computer Program as a teaching tool.

The ST-5 Computer Program

HTRI's ST-5 Computer Program is the industry standard for single-phase heat exchanger design and rating software. It has been used for over thirty years to successfully design and rate heat exchangers in the process industry. Until now, the sophisticated technology used by ST was only available to HTRI members. That has changed with the availability of the ST Educational Version.

Major Features of the Program

Features of the ST Educational Version include:

- The same calculational power as the commercial version.
- HTRI's complete Stream Analysis Method for shellside single-phase heat transfer and pressure drop on plain and low-finned tubes.
- HTRI's proprietary methods for tubeside single-phase heat transfer in the laminar, transition, and turbulent regions.
- Full support for the most common shell, TEMA E class R.
- The ability for students to control the leakage streams present in the heat exchanger which permits them to determine the effect from fouling and other deposits.
- A "construction element" input selection that gives students experience with how heat exchanger elements are normally specified in industry.
- The ability to quickly and easily investigate the effect of construction elements on the overall performance.
- A graphical user interface for input data panels with "point-and-click" selection.
- Full graphical, context-sensitive help

In the Classroom HTRI's ST Educational Version will enhance every students' curriculum in heat transfer. The material supplied includes a User's Guide that serves as an introduction to heat exchanger technology and can be used for classroom instruction and in design lab courses where students are expected to learn by doing. The Educational Version may be used:

- In process heat transfer courses to explain the design of heat exchangers and how construction elements affect performance
- In undergraduate heat and mass transfer courses to give an introduction to heat exchanger technology
- In process design projects to make accurate calculations for heat exchangers in the process flow sheet
- In unit operations causes in general to provide a quick and relevant introduction to heat exchangers
- In a short course on heat exchanger design Most importantly, students will have a chance to use up to date industry technology prior to graduation.

Computer Requirements

The ST Educational Version runs on an IBM PC with the following requirements:

- Microprocessor: 80386 with 80387, 80486DX, or Pentium
- 4 M-bytes RAM
- Hard disk drive with 4 M-bytes available storage
- MS-DOS or PC-DOS version 5.0 or later. Will run on DOS shell under Windows.

About Heat Transfer Research, Inc.

Heat Transfer Research, Inc. is an international research consortium providing advanced research and software tools in the areas of heat transfer and fluid

dynamics. Established in 1962, HTRI products are currently used by more than 400 companies worldwide.

Ordering the Program

For more information on the ST Educational Version, including a special tutorial diskette, HTRI membership information, or to receive our newsletter, please write to:

Heat Transfer Research, Inc.
ST Educational Version
1500 Research Parkway, Suite 100
College Station, Texas 77840
USA

FAX: 409-260-6249
PHONE: 409-260-6200
E-Mail: EMJ8871@ZEUS.TAMU.EDU

HTRI, ST, and ST Educational Version are trademarks of Heat Transfer Research, Inc. All other trademarks mentioned are property of their respective owners.

EURECHA Teaching Program Project

The EURECHA Teaching Program Project collects and distributes chemical engineering programs to teaching institutions. The modifications make the programs suitable for IBM-PC machines and can be used interactively.

Programs available from EURECHA Secretariat

1.	SOPS:	State oriented property system
2.	TACS:	Simulation of control systems
3.	ENISYN:	Synthesis of energy integrated distillation systems
4.	CHEQUUS:	Chemical equilibrium calculation
5.	KINET:	A kinetic examination, data acquisition and regression program
6.	SENSKIN:	Sensitivity analysis of chemical source terms in reactors
7.	UCTNET:	Analysis and design of heat exchanger networks
8.	B-CAD:	Design and operation of multiproduct batch plants
9.	RDC:	Design of rotating disc extractors
10.	PHYS:	Knowledge based system for the selection of thermodynamic models
11.	PROCEDE:	Flowsheet drawing program

For More Information

To place an order, please contact the EURECHA Secretariat at:

EURECHA Secretariat
ETH-Zentrum
Technisch-Chemisches Laboratorium
Universitätstr. 6
CH-8092 Zürich
Switzerland

Telefax: +41 1 252 0975

CACHE Catalog of Products

To order CACHE products, complete the order form on the back of this page and send with payment to:

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

FAX (512) 295-4498

If you would like to receive a current CACHE Products Catalog, send a message to <cache@uts.cc.utexas.edu>, call (512) 471-4933, or write to the address listed above.

STANDARD ORDER FORM

Description of Item	Quantity	Unit Price for Departments		Total
		Supporting	Non-Supporting	
AI Monograph - Volume 1		$20	$25	
AI Monograph - Volume 2		$20	$25	
AI Monograph - Volume 3		$20	$25	
AI Monograph - Volume 4		$20	$25	
AI Monograph Set		$50	$75	
AI Case Study - Volume 1		$10	$17	
AI Case Study - Volume 2		$10	$17	
AI Case Study - Volume 3		$10	$17	
AI Case Study Set		$20	$35	
Process Design Case Study - Volume 1		$15	$35	
Process Design Case Study - Volume 2		$15	$35	
Process Design Case Study - Volume 3		$15	$35	
Process Design Case Study - Volume 4		$15	$35	
Process Design Case Study - Volume 5		$20	$40	
Process Design Case Study - Volume 6		$55	$80	
Process Design Case Study - Volume 7		$20	$40	
Interactive Computer Modules • (per course) • (set of four)		 $35 $100	 $75 $200	
CACHE Tools [ ] 3 1/2" PC disk [ ] 3 1/2" mac disk		$160	$250	
GPSS • (license agreement must be signed first)		$25	$25	
PIP		$50	$75	
ChemSep Version 2.2 • (without documentation) • (with documentation) (license agreement must be signed first)		 $100 + annual $60 $135 + annual $60	 $115 + annual $75 $150 + annual $75	
PICLES Version 3.1 • (license agreement must be signed first)		$ 95 + annual $75	$115 + annual $95	
POLYMATH		$125 + annual $75	$150 + annual $100	
Purdue Laboratory Simulation Software (DOW, AMOCO, EASTMAN, MOBIL)		$225 per module plus annual $25	$225 per module plus annual $25	
CACHE 25th Anniversary CD-ROM • (purchased in bulk) • (purchased individually)		 $5/copy $10/copy	 $5/copy $10/copy	

NAME: _____

PHONE/EMAIL: _____

ADDRESS: _____

_____

_____

Make checks payable to:

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

(512) 471-4933
fax (512) 295-4498
email: cache@uts.cc.utexas.edu

Note: Overseas orders are sent surface at no charge. Airmail is extra.

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:

Tuskegee University
University of South Alabama
University of Arizona
California State Poly. Univ., Pomona
San Jose State University
University of California, Davis
University of California, Los Angeles
University of Southern California
Colorado School of Mines
Colorado State University
University of Connecticut
University of New Haven
University of Delaware
Florida Institute of Technology
University of Florida
Georgia Inst. of Technology
University of Idaho
Illinois Inst. of Technology
Northwestern University
University of Illinois, Chicago
University of Illinois, Urbana
Rose-Hulman Institute of Technology
Tri-State University
University of Kansas
University of Louisville Speed Scientific School
University of Kentucky
Louisiana Tech University
McNeese State University
Northeastern University
University of Massachusetts, Amherst
Worcester Polytechnic Institute
University of Minnesota
University of Missouri -Rolla
Washington University
University of Mississippi
University of New Hampshire
Princeton University
New Mexico State University
University of Nevada, Reno
Clarkson University
Manhattan College
Rensselaer Polytechnic Institute
State University of New York at Buffalo
Syracuse University
University of Rochester
North Carolina A&T State University
North Carolina State University of Raleigh
University of North Dakota
Ohio University
University of Akron
University of Cincinnati
University of Toledo
Oklahoma State University
University of Oklahoma
Oregon State University
Bucknell University
Carnegie-Mellon University
Lehigh University
The Pennsylvania State University
University of Pennsylvania
University of Puerto Rico
Clemson University
University of South Carolina
South Dakota School of Mines & Tech.
Vanderbilt University
Lamar University
Rice University
Texas A&M University
The University of Texas at Austin
Brigham Young University
University of Utah
University of Virginia
Virginia Polytechnic Instititute
University of Washington
West Virginia Inst. of Technology
West Virginia Graduate College
West Virginia University
University of New South Wales
University of Sydney
University of Newcastle
The University of Queensland
University of Adelaide
University of Alberta
University of Calgary
University of New Brunswick
Technical University of Nova Scotia
Lakehead University
McMaster University
Queen's University
Royal Military College of Canada
University of Toronto
University of Waterloo
University of Ottawa
University of Laval
Danmarks Tekniske HØjskole
Imperial College
University College London
University of Manchester Institute of Science & Technology
University of Bath
University of Bradford
Helsinki University of Technology
Lappeenranta Univeristy of Technology
University Dortmund
Universität Erlangen-Nürnberg
Universitaet Hannover
Indian Institute of Technology Kanpur
Cork Regional Technical College
Technion-Israel Institute of Technology
Yonsei University
Instituto Tecnologico y de Estudios Superiores de Monterrey
The Norwegian Institute of Technology
Universidade do Porto
Tiajin University
National Tsing Hua University
King Abdulaziz University
University of Maribor
University of Cape Town
Eidgenossiche Tech. Hochschule (ETH)
Middle East Technology University, Turkey

Topics in this issue of the CACHE Newsletter:

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

Nonprofit Organization
U.S. Postage Paid
Austin, TX
Permit No. 699