# CACHE NEWS

## News about Computers in Chemical Engineering Education

No. 57

Fall 2003

# TABLE OF CONTENTS

## CONFERENCES



## CACHE Standard Order Form

# 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 (Computer Aids for Chemical Engineering) Committee. The CACHE Committee was initially sponsored by the Commission on Education of the National Academy of Engineering and funded by the National Science Foundation. In 1975, after several successful projects had been completed, CACHE was incorporated as a not-for-profit corporation in Massachusetts to serve as the administrative umbrella for the consortium activities.

## CACHE ACTIVITIES

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

Individual inquiries should be addressed to:

CACHE Corporation
P. O. Box 7939
Austin, Texas 78713-7939
(512) 471-4933
FAX: (512) 295-4498
cache@uts.cc.utexas.edu

http://www.cache.org

**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 Christine Bailor with contributions from a number of CACHE members and representatives.

# CACHE Trustees

# Industrial Trustees

# CACHE Task Forces and Committees

## Standing Committees

**Publications**
Brice Carnahan **(Chair)**
Scott Fogler
Jeff Siirola

**Newsletter**
Scott Fogler
Peter Rony
Eddie Rosen

**Conferences**
Larry Biegler
Brice Carnahan
Peter Cummings
Tom Edgar
Gintaras Reklaitis
Warren Seider
Jeff Siirola **(Chair)**

**Industrial Trustees**
Rakesh Agrawal **(Chair)**
Edward Rosen
Jeff Siirola
Babatunde Ogunnaike

## Task Forces

**Software Engineering**
Tunde Ogunnaike **(co-chair)**
Ross Taylor **(co-chair)**

**Case Studies**
Larry Biegler
Gintaras Reklaitis

**Molecular Modeling**
Peter Cummings **(Chair)**
Scott Fogler
David Kofke
Warren Seider
Venkat
Venkatasubramanian
Phil Westmoreland

**Website**
Brice Carnahan
Peter Cummings
Michael Cutlip
James Davis
Frank Doyle
Thomas Edgar **(Chair)**
Scott Fogler
Peter Rony

# CACHE Industrial Affiliates

The following companies are members of the
CACHE Industrial Affiliates Program

**Aspen Technology**
Herbert Britt
herb_britt@aspentech.com

**Bayer Technology Services**
Thomas Daszkowski

**BP Chemicals**
Gordon Bentzen
bentzegw@bp.com

**Dow Chemical/Union Carbide**
Manuk Colakyan
colakymc@ucarb.com

**DuPont Central Research & Development**
Tony Liu
tony.w.liu@usa.dupont.com

**Eastman Chemical Company**
Jeff Siirola
siirola@eastman.com

**ExxonMobil**
Marco Duran
marco.a.duran@exxonmobil.com

**Fluent Inc.**
Richard LaRoche
richard.laroche@fluent.com

**ILOG Inc.**
Jim Claussen
jclaussen@ilog.com

**Lilly Research Laboratories**
Sangtae Kim
kim_sangtae@lilly.com

**Merck and Company Inc.**
Steve Simroth
steven_simroth@merck.com

**Mitsubishi Chemical Corporation**
Miho Hanafusa
2306475@cc.m-kagaku.co.jp

**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

Arizona State University
Auburn University
Brigham Young University
Bucknell University
California Institute of Technology
Carnegie-Mellon University
Case Western Reserve University
Christian Brothers University
Clarkson University
Clemson University
Cleveland State University
Colorado School of Mines
Colorado State University
Cork Institute of Technology
Cornell University
Curtin University of Technology
Dalhousie University, Canada
Darmstadt University of Technology
Ecole Polytechnic-University of Montreal
ETH Zentrum
Florida Institute of Technology
Georgia Institute of Technology
Howard University
Illinois Institute of Technology
Imperial College
Iowa State University of Science and Tech.
Johns Hopkins University
KAIST
Kansas State University
Kasetsart University
King Mongkut's Inst. of Tech. Thonburi
Lafayette College
Lakehead University
Lamar University
Loughborough University of Technology
Louisiana State University
Louisiana Tech University
Manhattan College
Massachusetts Institute of Technology
McGill University
McMaster University
McNeese State University
Michigan State University
Michigan Technological University
Mississippi State University
Monash University
Montana State University
New Jersey Institute of Technology
New Mexico State University
North Carolina State University of Raleigh
Norwegian Institute of Technology
Ohio State University
Ohio University
Oklahoma State University
Oregon State University
Polytechnic University
Princeton University
Purdue University
Queen’s University
Rensselaer Polytechnic Institute
Rice University
Rose-Hulman Institute of Technology
Rowan College
Royal Military College of Canada
State University of New York at Buffalo
Stevens Institute of Technology
South Dakota School of Mines & Tech.
Syracuse University
Technical University of Denmark
Technion-Israel Institute of Technology
Texas A&M University
Tufts University
Tulane University
Universidad Autonoma de Yucatan
Universidad de Concepcion
Universidade Tecnica de Lisboa
Universitat Dortmund
Universitat Kaiserslautern
Universitat Politecnica de Catalunya

**Chemical Engineering Departments**
**Supporting CACHE**

University of Adelaide
University of Akron
University of Alabama, Huntsville
University of Alabama, Tuscaloosa
University of Alberta
University of Arizona
University of Bath
University of Calgary
University of California, Berkeley
University of California, Los Angeles
University of California, Santa Barbara
University of Cape Town
University of Colorado
University of Connecticut
University of Dayton
University of Delaware
University of Florida
University of Illinois, Urbana
University of Iowa
University of Kansas
University of Kentucky
University of Laval
University of Louisiana at Lafayette
University of Louisville Speed Scientific School
University of Maribor
University of Maryland, College Park
University of Massachusetts, Amherst
University of Michigan
University of Minnesota, Duluth
University of Mississippi
University of Missouri, Rolla
University of Nevada, Reno
University of New Brunswick
University of New Hampshire
University of Notre Dame
University of Oklahoma
University of Pennsylvania
University of Pittsburgh
University of Queensland
University of Rochester
University of Saskatchewan
University of Sherbrooke
University of South Alabama
University of South Carolina
University of South Florida
University of Tennessee at Knoxville
University of Texas at Austin
University of Tokyo
University of Utah
University of Virginia
University of Washington
University of Waterloo
University of Wisconsin
Vanderbilt University
Villanova University
Wayne State University
Washington State University
Widener University
West Virginia Institute of Technology
West Virginia University
Youngstown State University

# Comments from the Editors

*Peter Rony, Ed Rosen, and Scott Fogler*

This is CACHE News issue No. 57 (Fall 2003), which means that we have been publishing CACHE News for at least 28 years . . . at the rate of two issues per year since 1974 or 1975. Our readers of the earliest issues of CACHE News have already retired.

If you glance at our current Table of Contents, you will observe that six of the twelve articles have been reprinted with permission from the American Society for Engineering Education (ASEE), and one article has been reprinted with permission from *Chemical Engineering Education*. Why reprint articles that have been published elsewhere? CACHE News is an informational newsletter. Reprints are made available so that those who missed the articles will have a second chance to read them on-line.

In this issue of CACHE News, we publish a nice mix of articles: several articles on computation in chemical engineering; application articles on VBA, GAMS, and FEMLAB; articles on statistics and web-based instructional tools in chemical engineering; an article on a virtual unit operations laboratory; and others.

CACHE Newsletter Committee

**Lakehead University**

I have developed software running under Solaris and Linux to help in the teaching of process control. The software called ColdSym brings up a graphic of a process in a "cold" state, that is, with nothing happening. The graphic includes valves and sensors which student can link using DCS-like block programming. The intent is to teach the structure of feedback loops as well as feedforward, cascade and constraint controls. Then the student can start up the process by turning on flows, steam, etc.

New simulations are created by preparing a process graphic; writing a C/C++ dynamic model which is compiled and linked with the ColdSym and various OpenGL libraries; and preparing a data file describing the process in its cold state.

I will be describing this software to the CSChE meeting in Hamilton in late October, and making the libraries and examples available on the web.

Allan Gilbert, PhD., P.Eng.
Chair, Department of Chemical Engineering
Lakehead University

**University of Massachusetts at Amherst**

Jeffrey M. Davis www.ecs.umass.edu/che/faculty/davis.html has joined the Chemical Engineering Department at University of Massachusetts, Amherst www.ecs.umass.edu/che as a new Assistant Professor. Jeff is a 2003 Ph.D chemical engineering graduate of Princeton with Professor Sandra M. Troian, www.princeton.edu/~stroian, and a 1999 B.S. graduate of MIT. His work in theoretical fluid mechanics focuses on microfluidics, as illustrated by his doctoral work on stability and control of thermally driven flow on micropatterned substrates.

Michael F. Malone, www.ecs.umass.edu/che/faculty/malone.htm has been named the Ronnie and Eugene Isenberg Distinguished Professor of Engineering at University of Massachusetts, Amherst, www.ecs.umass.edu/che. The purpose of this professorship is to enhance teaching and research in the combined fields of engineering and business and is awarded to a faculty member "who has demonstrated exceptional teaching and research skills and has achieved distinction in a specific area of engineering." Mike is the past department head and a nationally-known researcher. His work currently focuses on the computer-aided design and synthesis of separation systems used in the chemical and petrochemical industries. He is the author of 98 peer-reviewed publications, and co-author of two books. He has received a slate of prestigious honors, including a General Electric Outstanding Teaching Award, a University Distinguished Teaching Award, and Outstanding Senior Faculty Award from the College of Engineering, and the Computer and Systems Technology Division Award from the American Institute of Chemical Engineers. He is also Meeting Program Chair of the 2003 AIChE Annual Meeting in San Francisco.

Susan C. Roberts, www.ecs.umass.edu/che/faculty/roberts.html was named co-winner of the 2003 Outstanding Junior Faculty Award of the College of Engineering www.ecs.umass.edu/che at University of Massachusetts, Amherst.

A new engineering laboratory building, www.umass.edu/fp/projinfo/currproj/ecsc_ii/images.htm is nearing completion at University of Massachusetts, Amherst, to be shared by Civil and Environmental Engineering and the Chemical Engineering Department www.ecs.umass.edu/che. Occupancy is slated for March 2004.

**University of Texas at Austin**

We opened the new laboratories in Welch this quarter. Noteworthy is the fact that the College of Engineering contributed very significantly to the cost of the lab which happens to be located in Welch Hall, a College of Natural Sciences Space. The lab is used by Engineers every day but still, it is an unusual thing to see. Only at UT! I think it is Terrific. TEL also contributed a large sum of money toward the lab. There are pictures available...including the mariachi's who were there for the opening party and the Deans and President Faulkner.

I am proud to have received the Photopolymer Science Award in Japan this June and an announcement was just made that I won the Applied Polymer Science award from the American Chemical Society. This award will be given at the Spring meeting of the ACS

Professor Grant Willson
Department of Chemical Engineering
The University of Texas at Austin

# "Computing Through the Curriculum: An Integrated Approach for Chemical Engineering"

## CACHE Corporation (11/11/03)

contact: T.F. Edgar (edgar@che.utexas.edu)

**Abstract**

This white paper focuses on the integration of computing throughout the chemical engineering curriculum. The computing experience for undergraduates in chemical engineering should have continuity and be coordinated from course to course, because a single software solution is difficult to achieve in practice. This paper covers the topics of teaching computer programming, software selection, mathematical modeling instruction, and teaching process and product design. Appendices include a recent survey of computing practices in industry and descriptions of integrated computing approaches at selected departments.

**TABLE OF CONTENTS**

B.6 University of California, Santa Barbara

## 1.0 Computing and the Systems Approach in Chemical Engineering

The recent Frontiers in Chemical Engineering (see http://mit.edu) identified the systems area as one of the three key focus areas for the future chemical engineering curriculum. As stated in the report of the workshops, "the systems component of the curriculum equips the graduate to:

- create and understand mathematical descriptions of physical phenomena,
- scale variables and perform order-of-magnitude analyses,
- structure and solve complex problems,
- manage large amounts of messy data, including missing data and information,
- resolve complex and sometimes contradictory issues of process design: sensitivity of solutions to assumptions, uncertainty in data, what-if questions, and process optimization."

The systems component of the curriculum trains students in computational tools for synthesis, analysis, design, and manipulation of chemical and biological processes. In the systems approach students learn how to convert scientific facts and principles of chemical and biological systems into engineering decisions. This includes computer-based methods for dynamic and steady-state simulation at multiple length and time scales, statistical analysis of data, sensitivity analysis, optimization, parameter estimation and system identification, design and analysis of feedback systems, methods for monitoring and diagnosis, and methods for design of products and processes.

## 2.0 Teaching of Computer Programming and Computer-based Courses

### 2.1 Teaching Computer Programming

In the area of computing software, there is a noticeable disconnect between industry and academia. Appendix A presents a survey of computing practices of recent graduates in chemical engineering. Typically Chemical Engineering Departments teach the use of MATLAB, MathCAD, Mathematica, or Maple but not Excel. Yet in industry, Excel is the dominant computer package in use. There has been resistance to teaching Excel, because it is difficult to analyze the logic in Excel code. However, it is now possible to program spreadsheets using Visual Basic, where programming logic is more transparent.

Many departments no longer require a course in a computer programming language such as Fortran, C, or C++. It has been suggested that teaching computer programming is analogous to teaching plane geometry. It is a way of thinking but you may not have to use it. On the other hand, without some programming ability, engineers are limited by the built-in capability of commercial software without any way to extend it.

Many chemical engineering departments are wrestling with the following questions:

- When should computing be introduced to the chemical engineering student?
- How should computer programming in chemical engineering be taught, and how much formal programming instruction on languages such as C should be provided (vs. usage of computing tools such as MATLAB, spreadsheets, etc.)?
- Is a numerical methods course required and when does this occur in the course seqeunce? How many credit hours are needed?
- Should every course include some computing?

Since the mid-80's two approaches have been taken toward introductory computing for engineers. "CS 101" and the engineering tools approach. The CS 101 approach was catalyzed by the growth of computer science programs, which provided instruction in computer languages. The "CS 101" courses have migrated through several programming languages: Pascal, C, C++, and Java. In the engineering branch, software vehicles such as spreadsheets (first Lotus 123, then Quattro Pro, and now Excel), TK Solver, Mathcad, and MATLAB have gradually pushed out programming languages (primarily Fortran). These languages are becoming endangered species in these courses.

The "CS 101" branch would claim a number of reasons for existence (Clough, 2003):

- engineers should learn fundamental concepts of programming and computer science
- computing should be taught by computer scientists, not engineers
- engineering faculty are not interested in teaching computing languages to their students
- these courses provide a significant number of student credit hours (SCH) and budgetary resources

The "Engineering Tools Approach" branch believes:

- engineering students need a solid grounding in problem solving with modern computing tools
- engineering students need the knowledge and tools required in their professions
- engineering computing and problem-solving are best taught by engineers
- these courses comprise a significant number of student credit hours (SCH) and budgetary resources

An in-depth assessment reveals that the two branches are complementary and that many/most engineering students can benefit from both courses. However, most curricula are too congested to make room for both.

When it comes to computing needs, faculty often confuse what is important for their students with what is important for themselves. Faculty needs, more often than not, align with their research interests and activities, and these may be disconnected from the needs of their undergraduate students. Also, faculty often have a misinformed impression of the needs of professionals. Few discussions on computing needs proceed on the basis of evidence from alumni and employer surveys. Finally, computing is not part of the daily professional existence of most faculty and is not expected to be. Their computing skills can be oxidized, and most of their computing is carried out by their students.

The survey of industrial usage of computing in Appendix A indicates that a minority of recent graduates in chemical engineering actually perform programming on the job (although there is no clear definition of what constitutes programming in industry). The use of spreadsheets in chemical engineering practice is increasing. The application of spreadsheets in university courses are attractive because of student-driven usage. David Clough (Colorado) and Brice Carnahan (Michigan) have developed many examples of spreadsheet applications, as presented at the 2002 ASEE Chemical Engineering Summer School (see www.cache.org). In fact, it may be more appropriate to view computing as a subset of information technology for chemical engineers, which is the way industry views computing today. Teaching information technology through the curriculum may be the next incarnation of computing in chemical engineering.

### 2.2 Textbooks and Affiliated Software

The fragmented nature of software tied to leading undergraduate textbooks makes integration difficult, e.g.,

(a) material and energy balances: Felder and Rousseau – EZ Solve; Himmelblau and Riggs – POLYMATH
(b) thermodynamics: Sandler – MathCAD; Kyle – POLYMATH; Elliott and Lira – various programs
(c) separations: Wankat – Aspen
(d) process control: Seborg, Edgar, Mellichamp – MATLAB; Marlin – MATLAB; Bequette – MATLAB; Riggs – MATLAB/Excel; Control Station is a standalone package used at a number of departments
(e) chemical reaction engineering: Fogler – POLYMATH, Rawlings and Ekerdt – Octave
(f) product and process design: Seider, Seader, Lewin – Aspen, HYSYS, CHEMCAD, PROII

See the CACHE report (www.cache.org), "Chemical Engineering Problems with Solutions in MATLAB, POLYMATH, Excel, Mathematica, Maple, and MathCad" for a comparison of these different software packages applied to solving a number of prototypical problems in different courses in chemical engineering.

In addition to these courses, many departments are teaching a statistics course, which involves still one more software package such as JMP, SAS, or Minitab. Clearly using a subset of these textbooks sequentially through the sophomore, junior, and senior years will require a student to learn up to five or more different software packages.

Adding software packages outside of chemical engineering can push the total number of packages beyond ten. It would be desirable to keep the number of software packages below three or four if possible (note that Excel is not mentioned above). But usually textbooks are not chosen because of the bundled software. In addition, departments must address issues of software availability, licensing, cost, and providing software in computer labs vs. student-owned computers.

### 2.3 Faculty Control vs. Departmental Control of the Curriculum

One view of academia is that the professor is a "high priest" in his/her course with a large amount of discretion to select course content as well as the textbook to be used. In some departments it is unclear if the Department chair or curriculum committees have the authority to influence the content of the core courses. It is reasonable to allow an individual faculty member to have some flexibility in coverage of certain topics in a given course. However, because of the tight coupling of prerequisite courses, there should be some way of guaranteeing that knowledge, ability, and skills (ABET KAS's) of students who complete a given course are predictable. Most of the core curricula should be predictable but some curricula do not need to be that prescriptive. Even when KAS's are covered, 20% of a given course can be left to the discretion of the instructor. Another issue is whether computing should be incorporated into a given chemical engineering course and be covered regardless of which faculty member teaches the course. Many departments are discussing this issue but based on a limited sample, it appears that most departments have not reached a consensus on coordinating content or incorporating computing through the curriculum.

### 2.4 Teaching Process Simulators Through the Curriculum

It has been suggested that the difficulty of integration of computing tools mentioned above can be avoided by more extensive use of process simulators. At Virginia Tech, ChE undergraduates have been using Aspen Plus and Aspen Dynamics to solve problems in all subjects. It is fairly straightforward to convert a steady-state model in Aspen Plus into a dynamic model (with PID control schemes) in Aspen Dynamics. The applicability of Aspen Plus to mass and energy balances, thermodynamics (physical and thermodynamic property analysis, estimation and regression), multicomponent separations, reactor design and process flowsheet simulation are well-known. In process control Aspen Dynamics enables the students to evaluate controller tuning, process dynamics, startup and shutdown, etc. HYSYS has similar features and is used at many departments.

Recently, Version 2.0 of a CD-ROM, *Using Process Simulators in Chemical Engineering: A Multimedia Guide for the Core Curriculum* (Lewin et al., Wiley, 2003), has become available. Modules and tutorials are proved for self-paced instruction in the use of the process simulators to solve open-ended problems in courses on material and energy balances, thermodynamics, heat transfer, reactor design, separations, and product and process design. A 110-page document has been prepared for instructors suggesting the best instruction sequence and providing exercises and solutions, for each of the core courses (first introduced at the 2002 ASEE Chemical Engineering Summer School).

### 2.5 Numerical and Analytical Approaches in Modeling of Physical Behavior

Historically many engineering courses have previously been taught from an analytical viewpoint, but a transition is starting to occur, where numerical experiments are being gradually added. Problems and experiments should not be so simplified that they are not realistically formulated. Students are normally exposed to idealized fluid flow cases in the curriculum, for which application of theoretical concepts results in a solution of a one-dimensional ordinary differential equation or an algebraic equation. Therefore it is very easy for them to come away with the notion that theory is useless for most real-life situations.

Students should be able to select either analytical or numerical techniques to solve a problem, hence they should learn the advantages and disadvantages of either approach. Use of more sophisticated numerical tools such as CFD (computational fluid dynamics) will reduce the need to make many simplifying assumptions because you do not need as many assumptions to solve the problem numerically. Chemical Engineering students should understand that there are both numerical experiments and physical experiments. In some cases we can make observations from numerical experiments that you cannot see in physical data, but the converse is also true. This does not suggest that every experiment can be replaced with a simulator, but there should be a balance of the two approaches.

To prepare students for industrial practice, there should be a re-examination of the role of detailed analytical solutions. Is the purpose of some of these exercises the preparation of undergraduates for graduate school or industry? Today practicing engineers are not expected to carry out complex derivations in project work.

It is interesting to note that the public can run CFD simulators in science museums, therefore we ought to be able to teach this subject to chemical engineers. Once a fluid flow situation is analyzed theoretically or the governing principles are discussed, that same situation can be visualized using the computer. This visualization of the flow phenomena can significantly facilitate and enhance the learning process, especially for the visual learner. CFD software makes flow visualization easy. Students can simulate flow processes in a transient or steady state mode. Flow patterns can be displayed via velocity contours, velocity vector plots, or graphs of velocity profiles. A key element in flow visualization exercises is exploring the effect of different parameters. Using CFD, students can quickly change the size of the pipe, viscosity of the fluid, size of the particles, velocity of the feed, etc. and see the resulting changes in the flow behavior. This type of parametric analysis also ties in nicely with a discussion of dimensionless groups and geometric and dynamic similarity.

While computing and visualization can increase understanding, educators do not want students to view such simulators as black boxes. In the fluid mechanics course, simulations can become a mathematical exercise with little intuition, unless the instructor has the students solve a simple problem by hand first. More work on the software tools is needed, and it is critical to match the software tool to the student's knowledge base.

Two specific recent packages which have been developed are FlowLab (a finite volume-based code) by Fluent, Inc. and FEMLAB (a finite element-based code) by Comsol, Inc. FlowLab allows students to solve fluid dynamics problems without requiring a long training period. Using carefully constructed examples, FlowLab allows students to get started immediately without having to spend the large time commitment to learn geometry and mesh creation skills required by traditional CFD software. Current exercises developed includes sudden expansion in a pipe, flow and heat transfer in a pipe, flow around a cylinder, and flow over a heated plate, among others. In addition, professors can create their own examples or customize the pre-defined ones.

FEMLAB provides ready-to-use application modes, where the user can build his/her own model by defining the relevant physical quantities rather than the equations directly. The software also allows for equation-based modeling, which gives the user the freedom to create his/her equations. FEMLAB's programming language is an extension of the MATLAB language; this feature gives much flexibility to the user. FEMLAB's graphical interface includes functions for automatic mesh generation of a user-defined geometry. Recently a k-e turbulence model has been added to its menu of options.

Seider et al. (2004) present the design of configured consumer products, which usually involves two or three-dimensional simulations. In Chapter 19, Product Design momentum and species balances in a 2-D plasma CVD reactor are employed to produce thin Si films using CFD packages such as FEMLAB. This illustrates where it is very effective for students to use CFD packages to optimize designs – even without understanding all of the physical and chemical interactions in the transport-reaction processes.

Even with these recent advances in educational CFD software, this computing technology has been slow to penetrate undergraduate transport and reactor engineering courses. A recent CACHE Survey of all chemical engineering departments in the U.S. on barriers to implementing CFD identified a lack of knowledge concerning available CFD resources, a lack of professor training in CFD, the ease of use and the long learning curve associated with using CFD software in a given course, and cost of CFD software.

### 2.6 Laboratory Experiments Over the Internet

Web-access of laboratory experiments enables real chemical engineering laboratory equipment to be controlled and monitored interactively by computers that are connected to the Internet, i.e., under the command of users over the Web. This capability is now available in the labs at U. Tennessee-Chattanooga as well as other schools such as U.Texas-Austin, Columbia University, University of Toledo and MIT. This would permit faculty and students from any university to run Web-connected experiments at any time of the day or night, any day of the week. The laboratory station computer operates the equipment (pumps, valves, heaters, relays, etc.), collects the data (pressure, temperature, position, speed, concentration, etc.) and sends it to the web user. The U. Tennessee site is accessed through the web address, http://chem.engr.utc.edu/, and even includes audio and video of the operating equipment.

All established chemical engineering programs are facing increased financial pressure to keep existing laboratory experiments up-to-date and in satisfactory operating condition. Major operating costs of unit operations laboratories include maintenance and teaching assistant support. Using highly automated experiments for remote operations will allow a drastic reduction in TA time requirements for those particular experiments. In addition, by sharing the operation of the experiments among several universities, there can be a pro rata reduction in maintenance costs. There is also the opportunity to add experimental assignments to a lecture class using this technology. In a lecture class, it may be desirable to have students individually or in small groups carry out an experiment, much like a homework assignment; in contrast, a traditional experiment would require continuous supervision by teaching assistants (e.g., one week of TA time for an entire class). Therefore using an internet-based experiment can greatly reduce the time commitment by the TA. However, traditional experiments will remain in the curriculum to give students "hands-on" exposure, but they can be augmented with internet labs.

## 3.0 Process and Product Design

Historically there has been a process design emphasis in the curriculum that is now transitioning to a dual product and process design emphasis. This means that a framework is needed to make process decisions in order to make structured products. This has added a performance layer, i.e., not just purity of the product. Given a structure, we can often predict at some level what the properties of the material are likely to be. The accuracy of the results and the methods used to treat them depend critically on the complexity of the structure as well as the availability of information on similar structures. For example, various quantitative structure property relationship (QSPR) models are available for the prediction of polymer properties. However, the inverse engineering design problem, designing structures given a set of desired properties is far more difficult. The market may demand or need a new material with a specific set of properties, yet given the properties it is extremely difficult to know which monomers to put together to make a polymer and what molecular weight the polymer should have. Today the inverse design problem is attacked empirically by the synthetic chemist with his/her wealth of knowledge based on intuition and on experience. A significant amount of work is already under way to develop the "holy grail" of materials design, namely, effective and powerful reverse-engineering software to solve the problem of going backwards from a set of desired properties to realistic chemical structures and material morphologies that may have these properties. After this is completed, a subsequent step would involve how to manufacture the desired new product.

A chapter on Molecular Structure Design in Seider et al. (2004) contains simple optimization procedures using GAMS to determine polymer repeat units, refrigrants, and solvents that have desired properties using group-contribution methods. Eventually, these will be replaced (and augmented) by molecular models.

Another subject related to product design is the scheduling of batch processes, which can be done using simple simulation techniques, as in BATCH PLUS and SUPERPRO DESIGNER. Hence design of optimal processing can be viewed as "product design" for specialty chemicals. Clearly, spreadsheets and optimization packages can also be used for many of these computations. Finally, the use of large databases and software systems for equipment sizing and purchase and installation cost estimation, such as ASPEN IPE, are being commonly used throughout the chemical industries for product and process design.

### 3.1 Molecular Modeling

A molecular-level understanding of chemical manufacturing processes would greatly aid the development of steady-state and dynamic models of these processes. Process modeling is extensively practiced by the chemical industry in order to optimize chemical processes. However, one needs to be able to develop a model of the process and then predict not only thermochemical and thermophysical properties but also accurate rate constants as input data for the process simulation. Another critical set of data needed for the models are thermophysical properties. These include such simple quantities as boiling points and also more complex phenomena such as vapor/liquid equilibria phase diagrams, diffusion, liquid densities, and the prediction of critical points. A key role of computational chemistry is to provide input parameters of increasing accuracy and reliability to the process simulations.

Under the NSF grant, "World Wide Web-Based Modules for Introduction of Molecular Simulation into the Chemical Engineering Curriculum", seven university experts in molecular simulations have worked with the CACHE Molecular Modeling Task Force (MMTF) to develop WWW-based modules to facilitate introduction of molecular simulation into the chemical engineering undergraduate curriculum. These teaching modules can be integrated directly into chemical engineering core undergraduate courses, supplying for the instructor and the student the appropriate linkage material between macroscopic concepts currently taught in these courses and molecular simulations designed to aid student understanding of the molecular underpinnings of the phenomena. Modules are centered around Java Applets that run the molecular simulations and provide an "experimental" simulation platform for students to explore concepts. In addition, modules contain instructor materials, fundamental tutorials, student problems, and assessment materials.

MMTF has designed a consistent web-based interface that organizes all of the material in each module and developed scripts using *perl* that ease the job of putting the written material into this common format. The developer of a module must construct simple text files, perhaps with HTML markup that permits inclusion of figures and tables.

Then he or she runs the files through the *perl* script, which adds HTML formatting and links to put the set of files into the common configuration. The files are uploaded to the module site for anyone to access. This site is perhaps best accessed through the *Etomica site*. *Etomica* is a Java-based support environment developed for the modules project, which has now been expanded for other applications: go to http://www.ccr.buffalo.edu/eto and click on the "modules" link in the navigation bar on the left.

Following is a list of phenomena and concepts for which modules are completed or planned:

- Chemical reaction equilibrium
- Osmosis
- Diffusion
- Molecular dynamics
- Normal modes of a solid
- Chemical reaction kinetics
- Dissipative particle dynamics
- Surface tension
- Crystal viewer
- Joule-Thomson expansion
- Self assembly
- Chemical potential
- Multicomponent phase equilibrium
- Heat transfer
- Atomic billiards
- Viscosity

Contact David Kofke at the University of Buffalo for further information.

## 4.0 Conclusions and Recommendations

One way to foster renewal of the curriculum is to identify departments where curriculum revision is being carried out and to evaluate best computing practices and current trends. There may not be one answer because of different constraints various universities operate under, such as number of faculty in the department and whether computing courses are taught outside the department. Appendix B presents a number of different departmental approaches to integrating computing through the curriculum.

CACHE makes the following recommendations on computing through the curriculum:

(1) The "systems" component of chemical engineering is tied closely to computing tools, so strengthening the systems area as recommended by the Frontiers of Chemical Engineering Workshops should also strengthen the use of computing as well.

(2) There is increasing pressure on the total number of hours in the curriculum, especially with the addition of life science courses. Departments should continue to re-examine whether a formal three or four credit hour computer programming course is required for the chemical engineering degree (vs. teaching how to use software or write m-files in MATLAB, for example). The chemical engineering computing course also provides students with a valuable experience in quantitative problem-solving.

(3) The number of software tools that implement numerical methods used by students should be minimized; departmental agreement on software used in each course should be reached within the faculty. Faculty need to reach consensus on how student computing skills can grow systematically through evaluating each course in the curriculum.

(4) Courses such as transport phenomena and thermodynamics offer new possibilities for introduction of computing physical and chemical behavior, such as with computational fluid dynamics or molecular modeling. Process design can add a product design emphasis by using such tools as well.

(5) Internet-based laboratories offer a new means of bringing live experimental data into lecture-based courses, in order to reinforce theoretical concepts.

(6) To prepare students to optimize process designs, it helps to expose students to process simulators for solution of a problem(s) in the core courses of the chemical engineering curriculum. Also, as software develops and product design is added at the senior level, instructors must select from among optimization packages (such as GAMS), batch process simulators (such as BATCH PLUS and SUPERPRO DESIGNER), and packages for estimating equipment sizes and installation costs (such as Aspen IPE). The use of comprehensive software packages and databases is common in industrial design and needs to be introduced in design courses and utilized for solution of design projects.

**Acknowledgments**

Contributors to this paper include T.F. Edgar, W.D. Seider, D.E. Clough, J. Curtis, B.L. Knutson, P.R. Westmoreland, J.J. Siirola, Chau-Chyun Chen, G.V. Reklaitis, R. LaRoche, E.M. Rosen, D. Kofke, M. Cutlip, and M. Shacham.

**References**

Clough, D.E., “ChE’s Teaching Introductory Computing to ChE Students – A Modern Computing Course with Emphasis on Problem Solving and Programming”, *ASEE Annual Meeting*, 2002.

Lewin, D.R. et al., “Using Process Simulators in Chemical Engineering”, CD-Rom for Seider et al. (2004) textbook.

Seider, W.D., Seader, J.D., and Lewin, D.R., *Product and Process Design Principles*, Wiley, New York, 2004.

**Appendix A: Computing Practices in Industry: How Recent ChE Graduates Use Computing**

– Powerpoint Presentation

## Appendix B: Different ChE Departmental Approaches to Integration of Computing

B.1. University of Texas-Austin - Integration of Computing

During 2001-2002 The Department of Chemical Engineering at UT-Austin revamped the computing thread in its curriculum in order to strengthen student background in computing. This action was in response to student and faculty dissatisfaction with the depth and continuity of computer training over the four years of the program. The curriculum modifications included:

1. Adding a new ChE freshman course, ChE 210: Introduction to Computing, focusing on basics of computing, MATLAB, and Excel.

2. Changing ChE 448 to ChE 348 to focus on Numerical Methods in Chemical Engineering and Problem Solving, in the second semester – sophomore year.

3. Changing the existing junior lab course (3 credit hours to two two-credit hour courses: ChE253K: Introduction to Statistics and Data Analysis, and ChE 253M, Fundamental Measurements Laboratory.

In addition, reinforcement of the computing tools was implemented in core ChE courses by identifying prototype problems where these tools (mostly Excel and MATLAB) would be used. Table B.1 shows the layout of the required courses and where computing is employed.

Table B.1. Computing Tools Used in the ChE Curricula at UT-Austin

| Year | Computing Activities |
|---|---|
| Freshman | Introduction to MATLAB, Excel (second semester) |
| Sophomore | Material and Energy Balances (Excel)<br>Numerical Methods (MATLAB, Excel)<br>Transport Phenomena (Excel) |
| Junior | Thermodynamics (Excel)<br>Fluid Flow/Heat Transfer (MATLAB, CFD)<br>Statistics (JMP, Excel)<br>Separations (Aspen, Excel)<br>Measurements Laboratory (JMP, Excel) |
| Senior | Reactor Design (MATLAB, Excel)<br>Process Control (MATLAB, Excel)<br>Unit Operations Laboratory (data acquisition/control, Excel)<br>Process Design (Aspen, @Risk, Excel) |

The advantage of the proposed changes is that computing and numerical analysis are now spread uniformly over the first three years of the program, namely two hours on computer software tools in the second semester of the freshman year (ChE 210), three hours of numerical analysis in the second semester of the sophomore year (ChE 348), and two hours of statistics in the third year, increasing statistics instruction from one hour to two hours (ChE 253K). The doubling of coverage of statistics was in response to feedback from industry on the importance of statistics in chemical engineering practice. Note that even with all of these changes, there was no net increase in the number of credit hours for the degree.

The objectives of the two modified computing courses (210, 348) are as follows. Upon completion of ChE 210 students should:

- Understand basic computer architecture and internal number representation.
- Have an appreciation for limitations in numerical accuracy.
- Be able to construct plots, fit data, and build new functions using Microsoft Excel.
- Demonstrate ability to create complex programs in a programming environment such as MATLAB.

Upon completion of ChE 348 students should:

- Be able to identify and formulate methods to solve specific classes of numerical problems, including linear equations, nonlinear equations, numerical integration (quadrature), least-squares curve-fitting, minimization of functions, and differential equations.
- Understand how software can be used to solve each class of problem.
- Know limitations of each method.

Detailed outlines of the three new/revised courses (210, 348, 253K) are given below. For each topic, the number of one-hour lectures are given in parentheses.

### ChE 210 (Introduction to Computing)

1. Introduction to Computers (5)
   - History of computing devices
   - Modern computer architecture
   - Number representation and round-off
   - Internet, web, HTML

2. Spreadsheets (7)
   - Simple cell arithmetic
   - Plotting data – data visualization – good graphics
   - Solver
   - Visual Basic for applications

3. Programming Concepts (9)
   - Problem analysis and specification
   - Algorithms and control structures
   - Flow Charts and pseudocode
   - Sequential processing (order of precedence, arithmetic operations)
   - Selection structures (if-end, if-else-end, if-else if-else-end)
   - Repetition structures (for, while)
   - Comparison operators and Boolean expressions

4. MATLAB Programming (8)
   - Matrices and vectors
   - Plotting
   - Scripts
   - Functions
   - Selection, repetition and logicals

**ChE 348 (Numerical Analysis)**

1. Review of Program Organization and Structure (4)
   - Overview of course and MATLAB
   - Review of programming, control structures
   - Review of Taylor series
   - Errors, accuracy and stability

2. Matrices (3)
   - Elementary matrix-vector operations
   - Properties of matrix operations: eigenvalues, diagonalization
   - MATLAB operations

3. Linear equations (3)
   - Gaussian elimination/partial pivoting
   - Tridiagonal and band diagonal matrices

4. Single Nonlinear Equations (3)
   - Graphical solution
   - Newton, secant, Broyden methods

5. Multiple Nonlinear Equations (5)
   - Graphing zero contours

Newton's method, partial derivatives
MATLAB: fsolve
Example: Multiple reactions, CSTR

6. Differential Equations (7)
   Review of ODE's: linear vs. nonlinear ODE's, order of ODE's, linear first order ODE's (integrating factors), and solutions of second order ODE's.
   Quadrature: Simpson, Trapezoidal methods
   Numerical integration of initial value problems, Runge Kutta method
   Shooting methods

7. Multiple ODE's (3)
   Simplest method: solving reaction network problems with multiple reactions
   Connections to multiple algebraic equations (eigenvalues etc.)

8. PDE's (3)
   Parabolic PDE's (heat conduction problem)
   Other boundary conditions

9. Optimization (3)

As time permits, one or more of the following topics: Monte Carlo integration, molecular dynamics as an example of second order ODE's, stability and chaos

**ChE 253K (Introduction to Statistics and Data Analysis)**

1. Introduction (2)
   Discrete vs. continuous
   Variance of measurements
   Value of statistical analysis

2. Descriptive Statistics (3)
   Data sorting
   Frequency tables
   Stem and leaf plots
   Histograms
   Pareto plots
   Ogive plots

3. Probability (2)
   Defining probability
   Counting techniques, permutation and combination
   Additivity and Multiplicative rules
   Bayes' rule

4. Working with discrete random variables and probability distributions (2)
    - Define discrete random variables and continuous variables
    - Binomial distribution
    - Hypergeometric distribution
    - Poisson distribution

5. Working with Continuous Probability Distributions (2)
    - Normal distribution
    - Normal approximation to the binomial
    - Chi-Squared distribution

6. Functions of Random Variables (1)
    - Moments and moment generating functions

7. One and two Sample Estimations (3)
    - Statistical inference
    - Estimating the mean
    - Standard error
    - Tolerance limits
    - Estimating the difference between two means
    - Paired observations
    - Estimating variance
    - Estimating the ratio of variances

8. Hypothesis Testing (3)
    - Concepts
    - One and two tailed tests
    - Use of p-values
    - Choice of sample size
    - Tests of means
    - Tests of Variance

9. Linear Regression and Correlation (4)
    - Least square estimators
    - Analysis of variance approach
    - Linear regression case studies

10. Second Factorial Experiments (3)
    - Concepts of statistical experimental design and response surface analysis
    - Introduction to JMP for Box Behnken, etc.
    - Design of experiments and Response surface analysis with JMP

11. Statistical Process Control (2)

- Nature of control limits
- Purposes of control charts
- X-bar charts
- R-bar charts
- Cusum control charts

## B.2 Ben-Gurion University

Most students find it difficult to learn programming and in practice very few of them do any programming after finishing the basic course on the subject. After over 35 years that programming has been taught in the Chemical Engineering Departments, many schools now consider dropping those studies altogether because of their ineffectiveness.

At the Ben-Gurion University we have removed the required programming course from the curriculum two years ago and at the same time we extended an existing "Introduction to Personal Computers (IPC)" course by incorporating MATLAB programming into it. The main objective of the IPC course is to enable the students to solve engineering problems that involve complex consecutive calculations, solving linear and nonlinear algebraic and ordinary differential equations, and carrying out multiple linear and polynomial regression, using POLYMATH, Excel and MATLAB. To enhance the learning effectiveness the following advanced features are included in the course:

*Based on real life problems* – the study of each subject starts by presenting a real life problem, which requires numerical solution using one of the techniques mentioned above. Typical benchmark problems used include the calculation of thermodynamic properties using a cubic equation of state (Shacham et al, 2003), calculation of flow rate in draining a tank (Brauner and Shacham, 1994), calculation of adiabatic flame temperature (Shacham, 1998) and correlation of vapor pressure data using the Clapeyron, Antoine and Riedel's equations (Shacham et al, 1996).

*Multi-stage problems* – the problems require computer solution in several stages. Preliminary stages are solvable with the "user friendly" software packages (POLYMATH, Excel) while advanced stages require MATLAB programming (Shacham et al, 2003).

Programming by modification – solved examples are provided. The students are encourages to modify and extend the solved examples to solve their assignments.

*Self – grading* – students check and grade their own homework assignments and exams. The self-test program provides feedback on the errors made (Shacham, 1998). Most students will not be satisfied unless they get a working program that yields correct results.

*The problem-based approach* ensures that the students understand the need in learning programming and this increases their motivation. Using *multi-state problems* and *programming by modification* ensures that the students can get a working program in a reasonably length of time without losing faith in their programming abilities. The *self-grading* aspect provides an additional incentive to get a working program that yields correct results.

The final exam of the course in this last semester included the use of the "secant" method for solution of the adiabatic flame temperature problem (Shacham, 1998) with MATLAB programming. The students had an hour and a half to complete this assignment and 75% of them (40 students) managed to get a working program that yielded the correct solution. This clearly demonstrates the effectiveness of the teaching techniques that have been used in the IPC course.

**Introduction to Modeling and Computation for Chemical Engineers with POLYMATH, Excel and MATLAB**

**Chapter 1:** Complex Consecutive Calculations (pdf file, 233 KB)
Homework assignment **1:** Calculation of Molar Volume and Compressibility
Factor of a Gas Using the Redlich – Kwong Equation of State.
a. Problem Definition
b. Self Test

**Chapter 2:** Iterative Solution of a Nonlinear Equation (pdf file, 256 KB)
Homework assignment **2:** Calculation of Adiabatic Flame Temperature
a. Problem Definition
b. Self Test

Homework assignment **3:** Calculation of Molar Volume and Compressibility
Factor Using Various Equations of State.
a. Problem Definition
b. Self Test

Homework assignment **4:** Bubble Point and Dew Point for an Ideal Multi-component Mixture
a. Problem Definition
b. Self Test

**Chapter 3:** Matrix Operations and Solution of Systems of Linear Equations (pdf file, 193 KB)

**Chapter 4:** Multiple-Linear, Polynomial and Nonlinear Regression. Part I (pdf file, 215 KB), Part II (pdf file, 310 KB)
Homework assignment **5:** Fitting Correlations (Regression Models) to Vapor Pressure Data
a. Problem Definition
b. Self Test

Homework assignment **6:** Linearization of the Antoine Equation
a. Problem Definition
b. Self Test

**Chapter 5.** Introduction to Solution of Systems of Nonlinear Algebraic Equations

**Chapter 6.** Introduction to solution of Systems of Ordinary Differential Equations (pdf file, 263 KB)

## Comments, Corrections and Suggestions

Comments, corrections and suggestions regarding this course are welcome and should be sent to: shacham@bgumail.bgu.ac.il

## References (Section B.2)


Brauner, N. and M. Shacham, "Numerical Experiments in Fluid Mechanics with a Tank and Draining Pipe", *Comput. Appl. Eng. Educ*, **2**(3), 175-183 (1994).

Shacham, M., N. Brauner and M.B. Cutlip, "Replacing the Graph Paper by Interactive Software in Modeling and Analysis of Experimental Data", *Comput. Appl. Eng. Educ.*, **4**(3), 241-251 (1996).

Shacham, M., "Computer Based Exams in Undergraduate Engineering Courses", *Comput. Appl. Eng. Educ.*, **6**(3) 3, 210-209 (1998).

Shacham, M., N. Brauner and M.B. Cutlip, "An Exercise for Practicing Programming in the ChE Curriculum", *Chem. Eng. Educ.*, **37**(2), 148 (2003).

### Appendix B.3 University of Colorado

There are five goals in teaching computing to undergraduates in chemical engineering:

- fundamental knowledge of computing, programming and computers
- awareness of and preparation in emerging aspects of computing
- computing requirements in the other courses of their curriculum
- knowledge and skills required by engineers in their day-to-day professional lives
- opening the door for further study and specialization in computing and computer science

The argument for fundamental knowledge is sound. Such knowledge in computing will transcend the skills and tools of the day. Fundamentals provide the foundation. A criticism is that fundamentals tend to be abstract and difficult for students to grasp and appreciate. Most students learn better inductively, generalizing fundamentals from specific and practical exercises and examples.

Given the rapid rate of change in computing, teaching only the state of the art in the profession will leave students out of date by the time they get there. Therefore, educating them in emerging trends is important. A problem with this is judging which trends will stick and which will be flashes in the pan.

Students will appreciate and be motivated by the acquisition of skills that they can put to immediate use, even during the semester in which they are taking the introductory computing course. Of course, focusing too much on immediate needs may miss the mark when it comes to professional needs. Many computing tools used in the curriculum satisfy learning objectives but are of little use in professional life.

Teaching students in the context of computing vehicles used by practicing professionals has attractive payouts down the road. Focusing on the day-to-day problem solving activities of engineering professionals has high relevance and importance. However, there are difficulties. The learning curve for some software packages is far too steep, and some packages have a knowledge prerequisite, especially in mathematics, that far outstrips the abilities of first-year students.

There is also a significant problem with using tools that automate tasks, where the user has little view of the internal operations of the task – the *black-box* syndrome. This works fine and is greatly appreciated by the professional that already has knowledge of and experience with the task being performed. But there is a great temptation for mindless button-pushing (or mouse-clicking) by students who should be learning what is behind the button. Also, there is the danger of becoming trapped by the built-in capabilities of one's software, in other words, being incapable of extending the software capabilities through programming.

A few students will want to specialize in computing. For example, some students will become software developers in the context of engineering applications. These individuals will require more education in computing, perhaps a minor or even a double degree. The introductory computing course in engineering should not attempt to redirect these students away from computer science; rather, it should open the door.

Since valid arguments can be made for the five areas of need listed above, it becomes a challenge to design an introductory computing experience that balances and, at least in part, satisfies the needs. We have attempted to meet that challenge at the University of Colorado.

In engineering at the University of Colorado, introductory computing is taught under an umbrella course number (GEEN 1300 Introduction to Engineering Computing, 3 credit hours). The various engineering degree programs (chemical, mechanical, civil, architectural, environmental, aerospace) each teach a section of this course, and these sections take on “flavors” according to the preferences of the particular program. Students in electrical engineering, computer engineering, and computer science do not take this course, rather a typical “CS101” course based on C/C++. A significant fraction of the entering students, typically 30%, are “open option,” not having declared an engineering major. These students are included in the sections of the GEEN 1300 based on their interest in and leanings toward an engineering major. Also, there has occasionally been an additional section of the course for “open option” students and students not yet in the College of Engineering.

Two years ago, ChE at Colorado initiated a change in this course, taking it away from its traditional Fortran/Excel base. In this transition, two central themes were preserved: scientific/engineering problem solving and structured programming. The new course is divided into four roughly-equal parts. First comes a segment on engineering problem solving using the Mathcad package. This is followed by a segment on engineering problem solving and elementary numerical methods with Excel. The third segment expands the use of Excel by introducing structured programming via its Visual Basic for Applications (VBA) language. And the final segment continues the themes with the Matlab package along with an introduction to vector/matrix calculations. Problems from chemistry, physics, engineering and ChE are used throughout.

There are some pedagogical keys to the success of this course. The combined use of Excel and VBA has a strong supporting case. Providing students with knowledge and skills that they can use immediately, during the same semester, in other courses and activities is important to student motivation. Providing a gateway to subsequent use of the software tools and, for some students, to building their computing knowledge in follow-on courses completes the picture.

The introductory engineering computing course has been established with a set of objectives that include:

1. **Problem Solving**
   - Apply the “engineering method” to the solution of quantitative problems
   - Evaluate engineering formulas, carrying units and appropriate precision through calculations
   - Practice working in groups to tackle large-scale engineering problems

2. **Symbolic Computing**
   - Enter and edit symbolic expressions in computer software
   - Manipulate and solve algebraic expressions
   - Carry out symbolic manipulations for calculus

3. **Spreadsheet Techniques**
   - Develop efficient spreadsheet skills
   - Set up and interpret “what-if” and case study scenarios
   - Organize and layout spreadsheet solutions to engineering problems

4. **Programming Fundamentals**
   - Learn how information is represented by different data types
   - Learn program-flow algorithm structure and modularity
   - Program with object-oriented features

5. **Elementary Numerical and Statistical Methods**
   - Develop the ability to solve single nonlinear algebraic equations using elementary numerical methods, such as bisection, false position or Newton’s method
   - Solve sets of linear and nonlinear algebraic equations
   - Carry out regression calculations

6. **Software Tools**
   - Develop skills with and knowledge of the following software tools:
     Mathcad 2001
     Excel 2000 & Visual Basic for Applications (VBA)
     Mathlab 6

The “Problem Solving” objective is a carryover from the old “slide rule” courses. Most entering students lack practice and abilities in numeric problem solving. Many of the lessons from the old courses still have much value and prepare students for the activities in their other courses, in particular, the ChE material & energy balances course. Achieving this objective has an obvious beneficial long-term impact.

“Symbolic Computing” is of immediate utility in the students’ math courses. They also make use of this in their science and engineering courses. It is common for students to check their manual work using the computer. A byproduct lesson learned is that not all equations or systems of equations have analytical solutions [not obvious to many freshmen].

"Spreadsheet Techniques" provides a problem-solving methodology that has the broadest and longest impact of any objective in the course. Excel is the day-to-day problem solving tool of most practicing ChE's, and it is the software tool used most frequently by ChE students. Spreadsheet methods are becoming recognized in their own right along with the need to teach them separately to ChE students. The author's AIChE short course in spreadsheet problem-solving has been one of the most frequently offered courses (approaching 100 offerings) over the past dozen years.

"Programming Fundamentals" represents the *lost objective* in many engineering computing courses. It has been retained in this course in a creative way. The fundamentals of structured, algorithmic programming and data structure are introduced via the VBA language within Excel. This provides a natural setting for students to "elevate" prototypes developed on the spreadsheet into more elegant and efficient VBA macros (Subs) and user-defined functions. The portability of the programming concepts is further emphasized by learning the m-script language in Matlab. Achieving this objective opens the door for students in two important ways:

- Students who move on to take the computer science courses have a leg up on students with no background in programming – our students enjoy greater success in these follow-on courses

- Programming opens the door to extension of many software packages – without the ability to program, users are forever limited to the conventional, built-in capabilities of the packages.

There are certain numerical and statistical methods that represent the bread-and-butter of applications in engineering, both in the academic curriculum and in practice. Several of these are within reach of entering students, and these are represented in the "Elementary Numerical and Statistical Methods" objective. Apart from the practical relevance of equation-solving and regression methods, the lessons learned through a disciplined approach to Gaussian elimination are of great general value.

Our students are exposed to a great variety of software tools during their undergraduate ChE curriculum [Word, Powerpoint, Excel, Mathcad, Matlab, Mathematica, Simulink, Polymath, EZ-Solve, HYSYS, Aspen+, Minitab, Control Station, LabView, LadSim, AutoCAD, to name a few]. To achieve the "Software Tools" objective, we choose to expose the students to several software tools that will be of general utility, teach portable concepts, be accessible and easily acquired, be relevant to their other courses and/or to professional practice. Additionally, we need to prepare the students for the onslaught of the other software packages they will encounter. They obviously need to become skilled at picking up new tools quickly.

The course objectives are embodied in a course outline as follows:

| Topic | No. of lectures | No. of Labs |
|---|---|---|
| Introduction, problem solving, MathCAD | 6 | 3 |
| Spreadsheet problem solving, Excel | 7 | 4 |
| Introduction to programming, VBA | 7 | 4 |
| Numerical Methods, MATLAB | 7 | 4 |
| | 27 | 15 |

The introductory engineering computing course is taught in several sections that align with the engineering disciplines and the instructors are faculty from those disciplines. Consequently, there is a limited emphasis on examples and problem solving within the particular discipline of the instructor and section. However, the course sections are similar enough that students can cross over sections. This is also important for engineering students who have not declared a specific major yet.

Most believe that a “learn by doing” approach must be an integral part to an introductory computing course. The GEEN 1300 course incorporates a lab component, replacing a 1-hour lecture meeting with a 2-hour workshop in a computer laboratory. The workshops are tutorial in nature with some open-ended, exploratory content. Lab sections are mentored by upper-class undergraduate students who were successful in the course as freshmen. Experience over the years has taught us that this works better than instruction by graduate student TAs, many of whom come to us with varied and limited computing experience.

Outside work is dominated by weekly computing projects that require deliverables of computer files and written reports. There are frequent in-class demonstrations in lieu of conventional lecture. All lecture materials are available to students before class time as Powerpoint files.

Nearly all students have their own computers in their dorm rooms, and, although University computer labs are available in many locations for their use around the clock, students prefer to do their wok on their own computers. For about 50% of the course, this need is easily answered by Excel, a standard package on their computers. Many students elect to acquire Mathcad for a student price of about $125. Fewer choose to buy the student edition of MATLAB, although many do this later in their academic careers when the software package comes into more frequent use.

From our alumni and employer surveys, we find that Matchcad and MATLAB are not generally available to practicing ChE’s. Of course, Excel is available to all. So, the former packages answer mainly educational and academic needs.

**Learning Goals (GEEN 1300)**

1. **Problem Solving**
   - Learn to apply the "engineering method" to the solution of quantitative problems
   - Develop the ability to evaluate engineering formulas, carrying units and appropriate precision through calculations

2. **Symbolic Computing**
   - Develop the skills to enter and edit symbolic expressions in computer software
   - Learn to use computer software to manipulate and solve algebraic expressions
   - Learn to carry out symbolic manipulations for calculus

3. **Spreadsheet Techniques**
   - Develop efficient spreadsheet skills
   - Learn to set up and interpret "what-if" and case study scenarios
   - Learn to organize and layout spreadsheet solutions to engineering problems

4. **Programming Fundamentals**
   - Learn how information is represented by different data types
   - Learn program-flow algorithm structure and modularity
   - Learn to use features of object-oriented programming

5. **Elementary Numerical and Statistical Methods**
   - Develop the ability to solve single nonlinear algebraic equations using elementary numerical methods, such as bisection, false position or Newton's method
   - Learn to solve sets of linear and nonlinear algebraic equations
   - Learn to carry out regression calculations

6. **Software Tools**
   - Develop skills with and knowledge of the following software tools:
     Mathcad 11
     Excel 2002 and Visual Basic for Applications (VBA)
     Matlab 6.5

The engineering computing course in ChE at Colorado introduces students to various bread-and-butter numerical methods. This is done in a "crawl-then-walk-then-run" fashion where students first solve problems with pencil and paper ("crawl"), then use the spreadsheet environment ("walk"), then program the method in VBA and later in MATLAB ("run"), and finally use the "black box" capabilities of the software packages (Mathcad, Excel and MATLAB). This approach has two benefits:

- students gain an appreciation and understanding of the method and its limitations that is not possible by merely pushing the buttons of the "black box" software features, and
- students learn the discipline required to understand and carry out algorithmic numerical method.

Methods taught include equation solving (single nonlinear equations, methods such as bisection, false position and Newton's), solving sets of linear algebraic equations via Gaussian elimination, and linear regression.

**Learning Goals for CHEN 4580 – Numerical Methods for Process Simulation**

1. **Macroscopic Conservation Laws**
   - Understanding of general conservation laws
   - Ability to write unsteady-state conservation laws that include the accumulation term omitted in steady-state problems
   - Recognition of ordinary differential equations with straightforward analytical solutions

2. **Algebraic Equations**
   - Ability to solve single non-linear equations
   - Understanding of solution methods for systems of non-linear equations and ability to implement those methods in a modern computational environment
   - Understanding of how problem structure can be used to improve solution efficiency
   - Recognition of common chemical engineering applications leading to algebraic equations

3. **Ordinary Differential Equations**
   - Knowledge and use of single-step and multi-step methods for solution of initial value ordinary differential equation problems
   - Understanding of what a "stiff" differential equation is and appreciation for how stiff equations can be handled differently
   - Familiarity with available software for solving ordinary differential equations
   - Recognition of common chemical engineering applications leading to ordinary differential equations

4. **Microscopic Balances**
   - Ability to use a shell balance to derive a local, differential conservation equation
   - Recognition of common chemical engineering applications involving microscopic balances

5. **Split Boundary Value Problems**
   - Understanding and use of numerical methods for split boundary value problems

- Familiarity with available software for split boundary value problems
- Recognition of common chemical engineering applications leading to split boundary value problems

**Catalog Description**

The use of macroscopic and microscopic balances for development of mathematical models to describe common chemical engineering unit operations; numerical methods for solution of model equations. Prerequisites CHEN 3210, and 3220.

The third course in the computing sequence at U. Colorado is "Applied Data Analysis". The learning goals are given below.

**Learning Goals (CHEN 3010)**

**1. Measurement Fundamentals and Error Analysis**
- Understanding of measurement principles including accuracy and precision
- Ability to assess and manage experimental error in engineering calculations
- Knowledge of measurement techniques for common variables, such as pressure, flow and temperature

**2. Characteristics of Experimental Data**
- knowledge of probability concepts that relate to experimental measurements
- ability to describe measurements using common distributions
- ability to represent distributions using standard graphical and computing techniques

**3. Statistical Methods**
- understanding of the relationship of sample statistics to background distributions
- ability to compute sample statistics and confidence intervals
- ability to apply hypothesis tests common to engineering analyses

**4. Model Building**
- understanding of the concepts of regression analysis, including correlation and analysis of residuals
- ability to compute linear, including multilinear and curvilinear, and nonlinear regression
- ability to express confidence intervals on model parameters
- ability to assess goodness of fit and discriminate amongst competing models

**5. Design of Experiments**
- understanding of factorial design of experiments and response surface methods
- ability to plan an efficient experimental campaign based on factorial design

- understanding of the principles of analysis of variance as they apply to factorial design
- ability to process and interpret the results of factorial experiments
- ability to develop response surface models from the results of factorial experiments and use these models for prediction and evaluation.

**Catalog Description**

Students learn to analyze and interpret data. Topics include typical engineering measurements, graphical presentation and numerical treatment of data, statistical inference, regression analysis, and design of experiments. Prerequisites are GEEN 1300 and APPM 2360.

### B.4. University of Kentucky

The incorporation of computing through the curriculum is being addressed at the University of Kentucky with a goal of achieving the following objectives:

- Student proficiency in computing skills representative of the breadth of computing tools encountered in industrial practice.
- Demonstration of student's ability to for in-depth analysis using selected individual computing tools.
- Demonstration of student confidence in their ability to employ computing tools and to learn new tools
- Cohesive and efficient approach to introducing and using computing tools through the curriculum.

These objectives are being achieved through changes in the delivery of current courses as well as proposed curriculum reform. Through surveys of current students, alumni, co-op employers, and our faculty, the department clearly identified deficiencies in our previous approach to introducing and using computing tools. Namely, our required programming language course (Fortran or C++), which is taught outside our department, was not preparing students for engineering analysis. In addition, the programming language was not being sufficiently utilized and reinforced in the subsequent core chemical engineering courses for which it was a prerequisite. Numerical methods packages were being utilized in core chemical engineering courses, but the use of multiple software packages (e.g, Maple, MATLAB and Mathematica) limited the students' perceived proficiency in the use of computing tools. Further, students were not well-prepared to use chemical engineering simulation packages (e.g., Aspen or Chemcad) at the level required in senior-level design projects. Finally, knowledge of Excel was required for a broad range of core classes but was not introduced in a systematic manner.

These survey results led the faculty to carefully evaluate the choice of computing tools and the implementation of these tools in our curriculum. The following steps have been undertaken to incorporate computing tools through our curriculum:

1. A clear description of the objectives relating to computing in our curriculum was developed (as stated above).
2. A limited number of computing tools were selected and a plan for their introduction and implementation in the curriculum was formulated.
3. The relevance of a programming language in our curriculum was formally assessed.
4. The development of a new computing tools course within our chemical engineering curriculum was examined.

As a department, we have chosen to focus on three computing tools (a numerical methods package, a chemical engineering simulation package, and a spreadsheet program) and the implementation of these tools through the curriculum (Table B.1). Maple was chosen over other commercial numerical packages such as MATLAB because it is widely available, sufficient for most undergraduate problems, and instills the basis for learning more powerful numerical packages. In addition to being used by our chemical engineering faculty, Maple is also introduced in the freshman and sophomore-level calculus courses at the University of Kentucky. Aspen has been our primary unit operations simulation package, and continues to be supported in recent textbooks examples. Computer-based tutorials (D.R. Lewin, W.D. Seider, J.D. Seader, E. Dassau, J. Golbert, D. Goldberg, M.J. Fucci, and R.B. Nathanson, "Using Process Simulators in Chemical Engineering: A Multimedia Guide for the Core Curriculum" Distributed by John Wiley & Sons, Inc,. Version 2) has greatly enhanced the ability to incorporate unit operations simulation packages into our core courses. Recognizing its importance in industrial practice and its practical value, Excel is now emphasized as a primary computing tool in our curriculum. Training in Excel in our unit operations laboratory is coupled with statistical analysis, thereby addressing two educational needs of our program. The overall goal of this implementation is to provide students with fewer computing tools, but to increase their proficiency and confidence in these tools by reinforcing their use throughout the curriculum. The exception to our "preferred list" of computing tools occurs in Process Control, where Simulink, a MATLAB tool, is a standard tool for the simulation and analysis of control systems.

Table B.1. Current implementation of computing tools at the University of Kentucky

| Year | 1st Semester | 2nd Semester |
|---|---|---|
| **Freshman** | Calculus I: Maple | Calculus II: Maple<br>CS Programming Language |
| **Sophomore** | Calculus III: Maple<br>CME Process Principles: ASPEN | Calculus IV: Maple<br>CME Thermodynamics: ASPEN |
| **Junior** | CME Separation Processes: ASPEN and Excel | CME Process Modeling: Maple |
| **Senior** | CME Unit Ops. Lab: Excel<br>CME Reactor Design: ASPEN<br>CME Process Design I: ASPEN and Excel | CME Process Control: Simulink<br>CME Process Design II: ASPEN and Excel |

Our current plan of implementation of computing tools reflects our department's choice to move away from a formal programming language course. Thus, we are piloting a 2 credit hour CME course (CME 1xx *Computational Tools for Chemical Engineers*, Spring 2004) aimed at freshmen-level students. Our purpose for the course is the meaningful introduction of computing tools (Excel, Maple, and ASPEN) which will be used throughout the curriculum. One goal of the new course is proficiency in Excel, including mastery of the basic interface, graphing, regression, and optimization. Knowledge of the tools of Maple is the goal for the second section of the course. We will focus on simple methods used to numerically solve problems (symbolic math with Maple is covered in our Calculus sequence) and avoid a "black box" treatment. In this freshman-level course, only a basic introduction of ASPEN for the solution of simple problems is an appropriate goal. In addition to developing proficiency with the user interfaces of these packages, the focus will be on numerical methods (Newton-Raphson, solving linear equation sets, numerical differentiation and integration) used across all packages. This course will incorporate *chemical engineering* problems that the students will see again in future core courses (e.g., flash calculations). *Computational Tools for Chemical Engineers* will be offered as an optional course during its pilot period, to be substituted for our 2 hour programming language course. As we move toward the full implementation of the freshman-level Computational Tools course as a required course, we anticipate that our integration of computing tools in our curriculum will be described by Table B.2. The placement of this course in the freshman year will allow us to increase our reliance on computing tools, particularly Excel, in the sophomore year.

Table B.2. Proposed implementation of computing tools at the University of Kentucky

| Year | 1st Semester | 2nd Semester |
|---|---|---|
| **Freshman** | Calculus I: Maple | Calculus II: Maple<br>CME Computational Tools: Maple, ASPEN, and Excel |
| **Sophomore** | Calculus III: Maple<br>CME Process Principles: ASPEN and Excel | Calculus IV: Maple<br>CME Thermodynamics: ASPEN and Excel |
| **Junior** | CME Separation Processes: ASPEN and Excel | CME Process Modeling: Maple |
| **Senior** | CME Unit Ops. Lab: Excel<br>CME Reactor Design: ASPEN<br>CME Process Design I: ASPEN and Excel | CME Process Control: Simulink<br>CME Process Design II: ASPEN and Excel |

### B.5. University of Massachusetts

see pdf file

**A FEMLAB Study:**
**The Effect of Heat Transfer on Flow Field At Low Reynolds Numbers In Vertical Tubes**

by

Edward M. Rosen
*EMR Technology Group*

## Introduction

The effect of heating or cooling a fluid at low Reynolds number flowing upward in a vertical tube has been studied by Hanratty, Rosen and Kabel (1) and by Scheele, Rosen and Hanratty (2). Rosen (3) used series representations of the velocity and temperature fields to present an approximate solution to the equations of motion and energy. The results were reported by Rosen and Hanratty (4).

In this study a numerical solution to the problem using finite elements is made using FEMLAB/Matlab (5, 6). The study is limited to the prediction of the point of flow inversion (null point) at the center of the tube during heating.

## Description of the Flow Field

Experiments (3) were carried out using water in a long vertical glass tube inside an outer glass tube in which heated water was circulated. A dye was used to observe the flow field. Water in laminar flow (parabolic velocity profile) entered the heated section.

If the dye flooded the field and then was allowed to be swept out with the clear entering flowing water, a long cigar shaped area of dye appeared in the center of the of the tube. Its beginning location (the inversion or null point) was a function of the flow rate of the water and the temperature difference between the entering water and the temperature of the heated portion of the tube. Figure (1) indicates the coordinate system and nomenclature used in the study.

At the start of the heated section the water at the inner tube wall is accelerated more than the water in the center of the tube. As a result the velocity profile begins to flatten out. If the temperature difference between the wall and fluid is large enough, the center line velocity continues to fall until there is a flow reversal. Farther downstream instabilities may cause the flow field to become turbulent.

It is the location downstream at which the center line velocity goes to zero that is the object of this study. It is desired to compare the FEMLAB solution to experiment.

## Equations Describing the Problem

FEMLAB (7) generally states the momentum (Navier-Stokes) and Energy Equations in in a vector form (steady state):

***Navier-Stokes*:**

$$- \nabla \cdot h (\nabla \mathbf{u} + (\nabla \mathbf{u})^{T}) + r (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{F}$$

The relationship of this equation to a 2D cartesian coordinate system is shown in Figure 2 (8). Figure 3 is a cylindrical coordinate system formulation (9).

The body force in the z direction (vertical) is a gravity term and is set to (the Boussinesq relationship):

Fz = alpha* g*rho0*(T-To)

where

alpha = coefficient of expansion
g = acceleration due to gravity
rho0 = density at the inlet conditions
T = fluid temperature
To = fluid temperature at inlet

***Energy Equation*:**

$$\nabla \cdot (-k \nabla T) = Q - r \, Cp \, \mathbf{u} \nabla T$$

This is the non-conservative form of the energy equation and implies that $\nabla \mathbf{u} = 0$ and is appropriate for incompressible fluids.

## Boundary Conditions

Four boundary conditions were set up:

1. The flow field is considered to be symmetric around the center line. Therefore there are no velocity components perpendicular to the center line.

2. The tube exit is considered to be at zero pressure.

3. The entrance velocity has a parabolic profile and the fluid (water) is at To.

4. Two types of boundary conditions were considered at the tube wall.

a. Constant Temperature set equal to Td. This neglects the resistance of the tube wall.

b. Specified flux. At the tube wall the the flux is

$$kc*( \partial T / \partial r)_{r=a} = ks * (Td-T)/(a \ln (d/a))$$

where

| | | |
|---|---|---|
| kc | = | thermal conductivity of fluid (water) |
| T | = | Temperature of fluid |
| r | = | radial distance |
| ks | = | thermal conductivity of tube (Pyrex) |
| Td | = | Temperature of the outer tube wall |
| a | = | inside tube radius |
| d | = | outside tube radius |

## Physical Properties

The physical properties of water are taken from (10). The properties are evaluated at the entrance temperature, To, by linear interpretation about the closest values found in the reference's tables. The data and calculations, given in Figures 4A and 4B generally reproduced those listed in (3). The coefficient of expansion (alpha) and temperature coefficient of viscosity (b) are taken from (3). Psi is calculated as a consistency check with (3).

## Setting up FEMLAB

The general model to be used is first specified in the Model Navigator – new File:

*Chemical Engineering Module*
*Axisymmetry*
*Momentum Balance*
*Navier-Stokes*
*Stationary*

Various choices are then made from the pull down menu:

*Draw*

The geometry of the long tube is represented as a long rectangle from the center to the inside radius of the tube.

An arbitrary rectangle is drawn (from the rectangle icon):

The rectangle is double clicked and a window appears. The length Y (z direction) is arbitrary but long enough to reflect the flow reversal point noted in the experiments.

The following values are entered in the window (In order to follow experiments more easily, actual dimensions are used rather than putting the equations in dimensionless form):

Xmin = 0.   Xmax = 0.01095 m (inside tube radius)
Ymin = 0.   Ymax = 1.0 m (this varies depending on the experiment)

The "zoom extents" button is clicked to center the long tube.

*Options*

*Add/Edit Constants* (Data for Run 24)

The physical properties of water for this run are given in Figures 4A and 4B. The "set" and "apply" buttons are clicked each time a value is entered. The Nomenclature indicates the general symbols used by FEMLAB and the symbols used in the simulation

| | | |
|---|---|---|
| Tin (same as To) | 300.983 | Kelvins |
| Cp | 4178.72 | J/(kg – K) |
| kc | 0.61898 | J/(s-m-K) |
| g | 9.814 | $m/s^2$ |
| To | 300.983 | Kelvins |
| Uin (same as $v_{max}$) | .0252601 | m/s |
| alpha | 0.000283914 | 1/Kelvins |
| rho0 | 996.309 | $kg/m^3$ |
| muo | 0.00083908 | kg/(m-s) |
| b | 0.0218595 | 1/Kelvins) |
| Td | 315.0944 | Kelvins |
| fact | 100. | This is a parameter that was varied to 50, 20, 10…1 in a series of runs to force convergence. |

*Subdomain*

The subdomain generally refers to the equations being solved. There is a subdomain for the Navier-Stokes equations and one for the Energy equation.

*Subdomain Settings* (Navier-Stokes)

*Coefficients* (FEMLAB symbol and simulation symbol used)

? = rho0 Density
? = mu Dynamic Viscosity
Fr = 0 Volume force r direction
Fz = FY Volume force z direction

*Options*

This accounts for the buoyancy force variation and the variation of viscosity with temperature. The values of ko and Cpo are constants:

*Add/Edit Expressions*
*Variable Name = FY*
*Add*
*Definition*
*alpha*g*rho0*(T-To)/fact*

*Variable Name = mu*
*Add*
*Definition*
*muo/(1+b*(T-To))*

*Boundary Settings*

| | | |
|---|---|---|
| 1 | LHS = slip | (Left Hand Side - center line symmetry) |
| 2 | v = Uin*(1-s^2) | (s is a parameter going from 0 to 1 at input)) |
| 3 | p = 0 | (exit pressure is zero) |
| 4 | RHS = no slip | (Right Hand Side - at the wall) |

*Multiphysics*

*Add/Edit Modes*
*Chem Axi, Convection-Conduction* (click on >>)
*Non-Consevative* (submode)

*Subdomain Settings (Convective-Conduction)* The general FEMLAB sysmbol is equated to the simulation symbol:

| | | |
|---|---|---|
| ? | = rho0 | Density |
| Cp | = Cp | Heat Capacity |
| k | = kc | Thermal Conductivity (Isotropic) |
| Q | = 0 | Heat source |
| u | = u | Radial Velocity |
| v | = v | z direction velocity |

*Init*
$T(t_o)$ = To

*Boundary Settings (Energy Equation)*

1 LHS = insulation
2 in: $T = T_{in}$
3 out: $q \cdot n$ = convective flux
4 RHS: for constant temperature boundary at the tube wall: = Td

for flux at the wall : = ks * (Td-T)/(a ln(d/a)) = 776.098 * (Td-T)

where:

ks = 1.12472 J/(s-m-K) from (3)
a = 0.01095 m
d = 0.0125 m

*Mesh*

*Initial*
*Refine *2*
*Refine Inlet and Outlet*

Alternate:

*Parameters*

*Max edge size, general (3e-3)*

*Solve*

*Solver Parameters*
*Nonlinear*
*Maximum Number of iterations 200*

*Post*
*Plot Parameters*
*ArrowLine*

*Geometry Boundaries*

C*ross section plot parameters*

*General*

*Line*

*Line*

The two point option allows the plotting of center line and radial temperatures and velocities.

## Executing FEMLAB

Execution of FEMLAB is started by clicking on *Solve*. There were three issues, however, that were needed to be resolved before successful convergence could take place.

1. *Memory*

   Initially 128MB memory was used on a 600 megahertz Intel Pentium III PC. For this problem 128 MB (or even 256 MB) seemed inadequate. A message "Out of memory" was often encountered. Upgrading memory to 512 MB allowed solutions to be obtained, though additional memory is clearly desirable for more detailed investigations.

2. *Mesh Size*

   Choosing an adequate mesh size is critical. Picking a large mesh size often leads to the message "Step Size Too Small" and the run is terminated. Choosing a small mesh size results in very long run times (hours). Selecting the initial mesh and refining twice is often adequate though trial and error may be required.

3. *Approach to the Solution*

   Attempting to solve both the momentum and energy equations at one time is often unsuccessful even if very long execution times are allowed. The most successful procedure is to:

   a. Solve the Navier-Stokes equation alone. This is done by:

      *Multiphysics*

      *Solve for variables*

      Highlighting Navier Stokes

   b. Once the Navier Stokes equation is solved then both the Navier-Stokes and the Energy Equation are solved together. This is done by

*Multiphysics*

*Solve for variables*

Highlighting both the Navier-Stokes and Energy Equation. To do this hold the *Ctrl* key down.

c. Modifying the volume force by the factor “fact” specified in:

*Options*

*Add/Edit Constants*

set fact to new value

The parameter fact is initially set to 100 and both equations are solved. Then the value of fact is reduced to 50, 20 and 10. Finally fact is set to 1 which is the solution to the problem..

**FEMLAB Results**

The value of z at which the center line velocity fell to zero is entered into the table of Figure 4B which then calculates the values of Gr, Re and Z* x 10^3 for plotting.

In all ten runs are selected from Reference (3). For each run FEMLAB simulations are made using a constant temperature wall boundary condition and for using a flux boundary condition which recognizes the resistance of the wall.

Figures 5 to 9 are plots taken from FEMLAB that illustrates the velocity and temperature profiles encountered for run 24. Similar profiles are encountered for both the constant temperature and flux runs.

Figure 10A summarizes all the FEMLAB simulations and compares the results to the experimental values given in (3). The data are plotted in Figure 10B in dimensionless form.

**Discussion of Results**

The plot of Figure 10B illustrates two trends:

1. Constant temperature boundary simulations fall below the null point experimental values.

   Experimentally, the greater the heating with a fixed flow, the closer to the entrance flow reversal takes place. In the experiments the wall temperature at the entrance and downstream must be less than Td due to the thermal resistance of the wall. Setting the wall temperature to Td in the simulations results in greater heating to the fluid causing the fluid to have flow reversal nearer the entrance. The FEMLAB simulations are therefore consistent with experimentally observed values.

Figure 11 is a plot of the wall temperature near the entrance for run 24. The wall temperature becomes steady (equal to Td) only after rising from an average of Td and To and going through an excursion. This is a result of the discontinuity.

2. Flux boundary condition simulations fall above the null point experimental values.

   Experimentally, the temperature of the fluid at the wall (at the entrance) is probably warmer than To (due to heat leakage). As a result the temperature of the fluid along the wall as it moves downstream will be warmer than the simulation which starts the fluid temperature at the wall at To.

   Figure 12 is a plot of the wall temperature for run 24 with a flux boundary. The boundary condition in the simulation forces the fluid temperature at the wall to be To even though the wall temperature is Td (less the resistance drop). Also, since the inlet fluid temperature is uniform the fluid flux must be initially zero at the wall. There is a discontinuity both in the temperature and the flux.

   Since the temperature of fluid along the wall at the entrance (and downstream) is less than the experimental conditions, the point of inversion in the simulation falls downstream of the experimental values (Figure 10B).

   It is unclear whether a finer mesh size would bring the simulated results closer to the experimental values. Memory limitations prevented testing this.

## Conclusions

FEMLAB has a wide range of capabilities, is very well supported and has extensive documentation, though this investigator found the documentation difficult to follow at times. FEMLAB gives quite reasonable results when compared to experiment. To get better simulations, however, finer mesh points may be needed.

## Nomenclature

*English*

| | |
|---|---|
| a | Inside tube radius |
| b | Temperature coefficient of viscosity |
| bo | Temperature coefficient of viscosity at To |

| | |
|---|---|
| alpha | Coefficient of expansion |
| Cp | Femlab Heat capacity |
| Cpo | Heat Capacity at To (used in simulation) |
| d | Outside radius of tube |
| F | Femlab vector of body forces |
| Fz | Body force in z direction |
| FY | Body force in z direction (usedin simulation) |
| g | acceleration of gravity |
| Gr | Grashof number = $a^3 r_o^2 g\ b_o$ (Td – To)/ $m_o^2$ |
| Gz | Graetz number = $p$ Re Pr/Z |
| k | Femlab Thermal conductivity |
| kc | Thermal conductivity of fluid at To (used in simulation) |
| ko | Equal to kc |
| ks | Thermal conductivity of glass tubing (used in simulation) |
| mu | Viscosity (used in simulation) |
| muo | Viscosity at To (used in simulation) |
| p | FEMLAB pressure |
| Pr | Prandtl Number = Cpo $m_o$ /kc |
| Psi | Parameter = bo (Td-To) |
| Q | FEMLAB heat flux |
| r | FEMLAB radial distance |
| rho0 | Density of water at To (used in simulation) |
| Re | Reynolds number = $a\ v_{avg}\ r_o / m_o$ |

| | |
|---|---|
| s | FEMLAB parameter $0 \le s \le 1$ |
| T, t | FEMLAB Temperature |
| Td, $t_d$ | Outside wall temperature (used in simulation) |
| To, $t_o$, Tin | Entering fluid temperature (used in simulation) |
| **u** | FEMLAB velocity vector |
| u | FEMLAB velocity in r direction |
| Uin | Equal to $v_{max}$ |
| v | FEMLAB velocity in z direction |
| $v_{avg}$ | Average z velocity in entering fluid (used in simulation) |
| $v_{max}$ | Maximum z velocity in entering fluid (used in simulation) |
| z | Distance downstream |
| Z | z/a |
| Z* | Z/( $p$ Re Pr ) |

*Other*

| | |
|---|---|
| $b$ | Same as alpha |
| $h$ | FEMLAB symbol for viscosity |
| $m$ | same as mu (viscosity) |
| $m_o$ | Viscosity at temperature To |
| $r_o$ | Density at temperature To |
| $\nabla$ | Vector differential operator (divergence) |

## References


1. Hanratty, T. J., Rosen, E. M. and Kabel, R. L., "Effect of Heat Transfer on Flow Field at Low Reynolds Numbers in Vertical Tubes", *Ind. Eng Chem*. 50, 815-20 (1958).

2. Sheele, G. F., Rosen, E. M. and Hanratty, T. J. , "Effect of natural Convection on Transition to Turbulence in Vertical Pipes"**,** *The Canadian Journal of Chemical Engineering*, June, 1960 pp 67-73.

3. Rosen, E. M., "Effect of Heat Transfer on Flow Field At Low Reynolds Numbers in Vertical Tubes" Ph. D. Thesis in Chemical Engineering, University of Illinois, 1959

4. Rosen, E. M. and Hanratty, T. J., "Use of Boundary-Layer Theory to Predict the Effect of Heat Transfer on the laminar-Flow Field in a Vertical Tube with a Constant Temperature Wall", *AIChE J*. Vol 7, No 1 , March 1961 p 112.

5. The MathWorks, Inc. 3 Apple Hill Drive, Natick, MA 01760-2098, http://www.mathworks.com

6. COMSOL,Inc. 8 New England, Executive Park, Burlington, MA 01803 http://www.comsol.com

**7.** FEMLAB, Chemical Engineering Module (Version 2.3) p 1-33.

**8.** Support@femlab.com

9. Hughes, W. F. and E. W. Gaylord, "Basic Equations of Engineering Science" Schaum Publishing Co., New York (1964)

10. Perry, J. H., "Chemical Engineer's Handbook**",** 3rd Edition McGraw Hill**,** New York, 1950

**Acknowledgement: This study was undertaken during a trial of FEMLAB/Matlab. The author wishes to thank MathWorks, Inc and Comsol, Inc for the use of their software.**

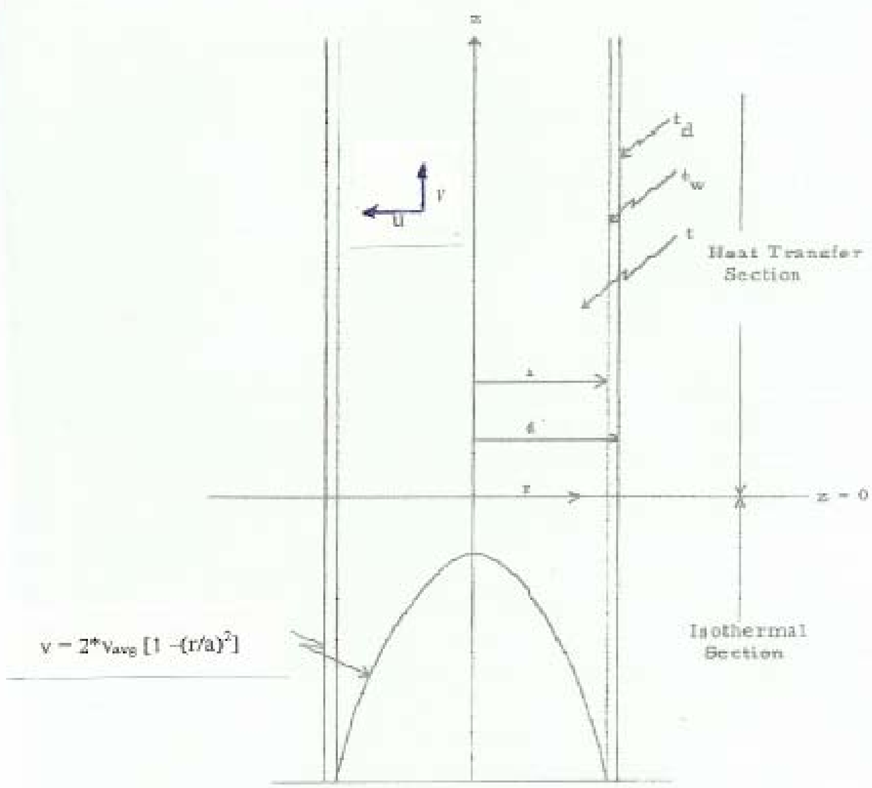



Figure 1. Definition of Coordinate System and Nomenclature

Consider the 2D column vector $\mathbf{u}$, spanning over the momentum equations in the x and y direction:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$$

The gradient will be the tensor

$$\nabla\mathbf{u} = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\ \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \end{bmatrix}$$

the transpose of which is

$$(\nabla\mathbf{u})^T = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial v}{\partial x} \\ \dfrac{\partial u}{\partial y} & \dfrac{\partial v}{\partial y} \end{bmatrix}$$

hence

$$\nabla\mathbf{u} + (\nabla\mathbf{u})^T = \begin{bmatrix} 2\dfrac{\partial u}{\partial x} & \dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} & 2\dfrac{\partial v}{\partial y} \end{bmatrix}$$

and

$$\nabla\cdot\eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) = \begin{bmatrix} \dfrac{\partial}{\partial x}\eta\left(2\dfrac{\partial u}{\partial x}\right) + \dfrac{\partial}{\partial y}\eta\left(\dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y}\right) \\ \dfrac{\partial}{\partial x}\eta\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) + \dfrac{\partial}{\partial y}\eta\left(2\dfrac{\partial v}{\partial y}\right) \end{bmatrix}$$

This expression can be identified with the viscous terms in the full momentum balance in eq. 1.58 in Hughes and Gaylord or eqns 3.2-17 to 3.2-19 in Bird, Stewart and Lightfoot, 1st ed.

For a Newtonian , incompressible fluid, the viscosity $\eta$ is constant, hence

$$\nabla\cdot\eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) = \eta\nabla\cdot\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) = \begin{bmatrix} \eta\left(2\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 v}{\partial x\partial y} + \dfrac{\partial^2 u}{\partial y^2}\right) \\ \eta\left( + \dfrac{\partial^2 u}{\partial x\partial y} + \dfrac{\partial^2 v}{\partial x^2} + 2\dfrac{\partial^2 v}{\partial y^2}\right) \end{bmatrix}$$

or reorganized

$$\nabla\cdot\eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) = \eta\begin{bmatrix} \dfrac{\partial}{\partial x}\left(\dfrac{\partial u}{\partial x} + \overbrace{\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}}^{=0}\right) + \dfrac{\partial^2 u}{\partial y^2} \\ \dfrac{\partial}{\partial y}\left(\underbrace{\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}}_{=0} + \dfrac{\partial v}{\partial y}\right) + \dfrac{\partial^2 v}{\partial x^2} \end{bmatrix} = \eta\begin{bmatrix} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} \\ \dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} \end{bmatrix} = \eta\nabla^2\mathbf{u}$$

Because $\partial u/\partial x + \partial v/\partial y = 0$ for an incompressible fluid. The above reasoning can be done for a 3D system as well.

Figure 2 Relationship to 2D Cartesian Coordinate System

$v_r$, $v_\theta$, and $v_z$ are the velocities in the $r$, $\theta$, and $z$ directions respectively. In the following section:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v_r \frac{\partial}{\partial r} + \frac{v_\theta}{r}\frac{\partial}{\partial \theta} + v_z \frac{\partial}{\partial z}$$

$$\begin{aligned}\rho\left[\frac{Dv_r}{Dt} - \frac{v_\theta^2}{r}\right] \\ = -\frac{\partial P}{\partial r} + F_r + 2\frac{\partial}{\partial r}\left(\mu\frac{\partial v_r}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\mu\left(\frac{1}{r}\frac{\partial v_r}{\partial \theta} + \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right)\right] \\ + \frac{\partial}{\partial z}\left[\mu\left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)\right] + \frac{2\mu}{r}\left(\frac{\partial v_r}{\partial r} - \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} - \frac{v_r}{r}\right)\end{aligned}$$

$$\begin{aligned}\rho\left[\frac{Dv_\theta}{Dt} + \frac{v_r v_\theta}{r}\right] \\ = -\frac{1}{r}\frac{\partial P}{\partial \theta} + F_\theta + \frac{2}{r}\frac{\partial}{\partial \theta}\left(\frac{\mu}{r}\frac{\partial v_\theta}{\partial \theta}\right) + \frac{\partial}{\partial z}\left[\mu\left(\frac{1}{r}\frac{\partial v_z}{\partial \theta} + \frac{\partial v_\theta}{\partial z}\right)\right] \\ + \frac{\partial}{\partial r}\left[\mu\left(\frac{1}{r}\frac{\partial v_r}{\partial \theta} + \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right)\right] + \frac{2\mu}{r}\left[\frac{1}{r}\frac{\partial v_r}{\partial \theta} + \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r}\right]\end{aligned}$$

$$\begin{aligned}\rho\frac{Dv_z}{Dt} \\ = -\frac{\partial P}{\partial z} + F_z + 2\frac{\partial}{\partial z}\left(\mu\frac{\partial v_z}{\partial z}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left[\mu r\left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)\right] \\ + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\mu\left(\frac{1}{r}\frac{\partial v_z}{\partial \theta} + \frac{\partial v_\theta}{\partial z}\right)\right]\end{aligned}$$

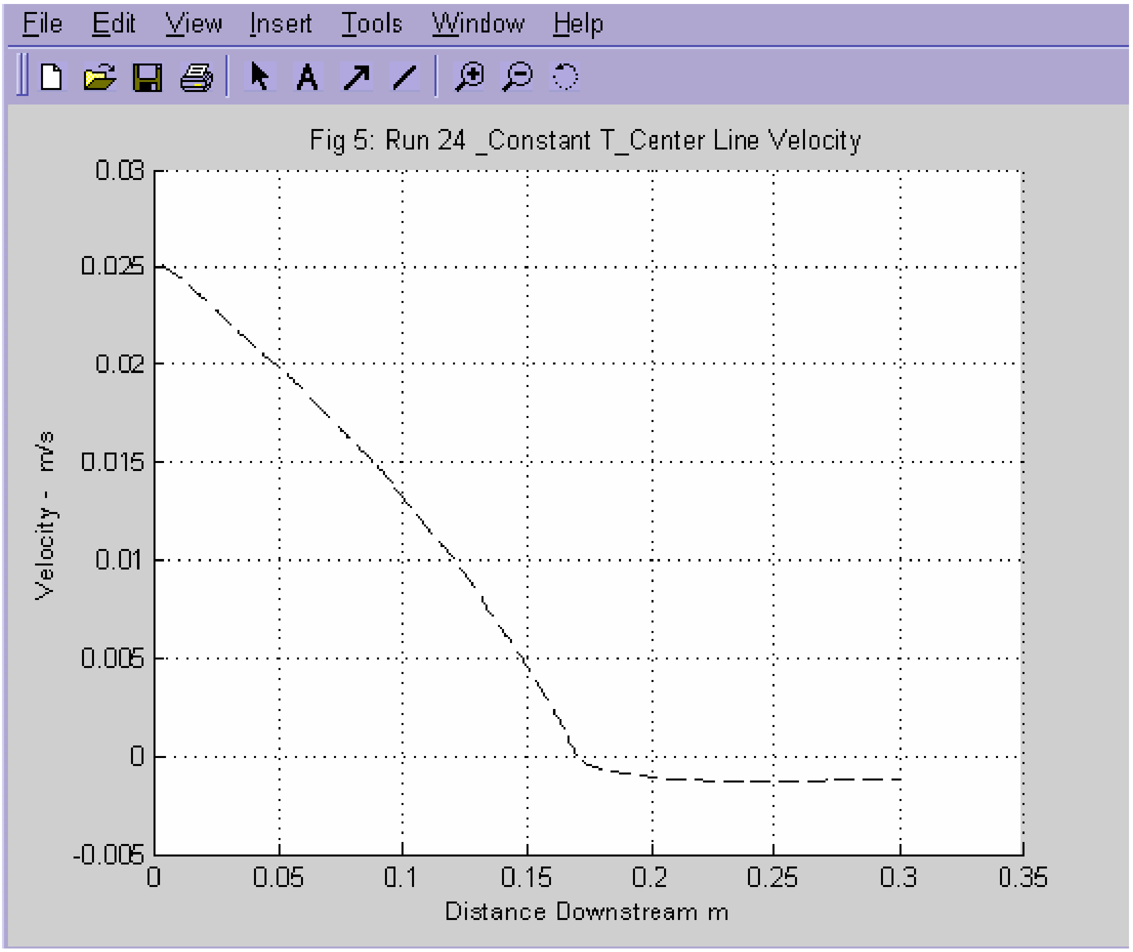
Figure 3. Navier-Stokes Equation of Motion for An Incompressible Fluid- Cylindrical Coordinates

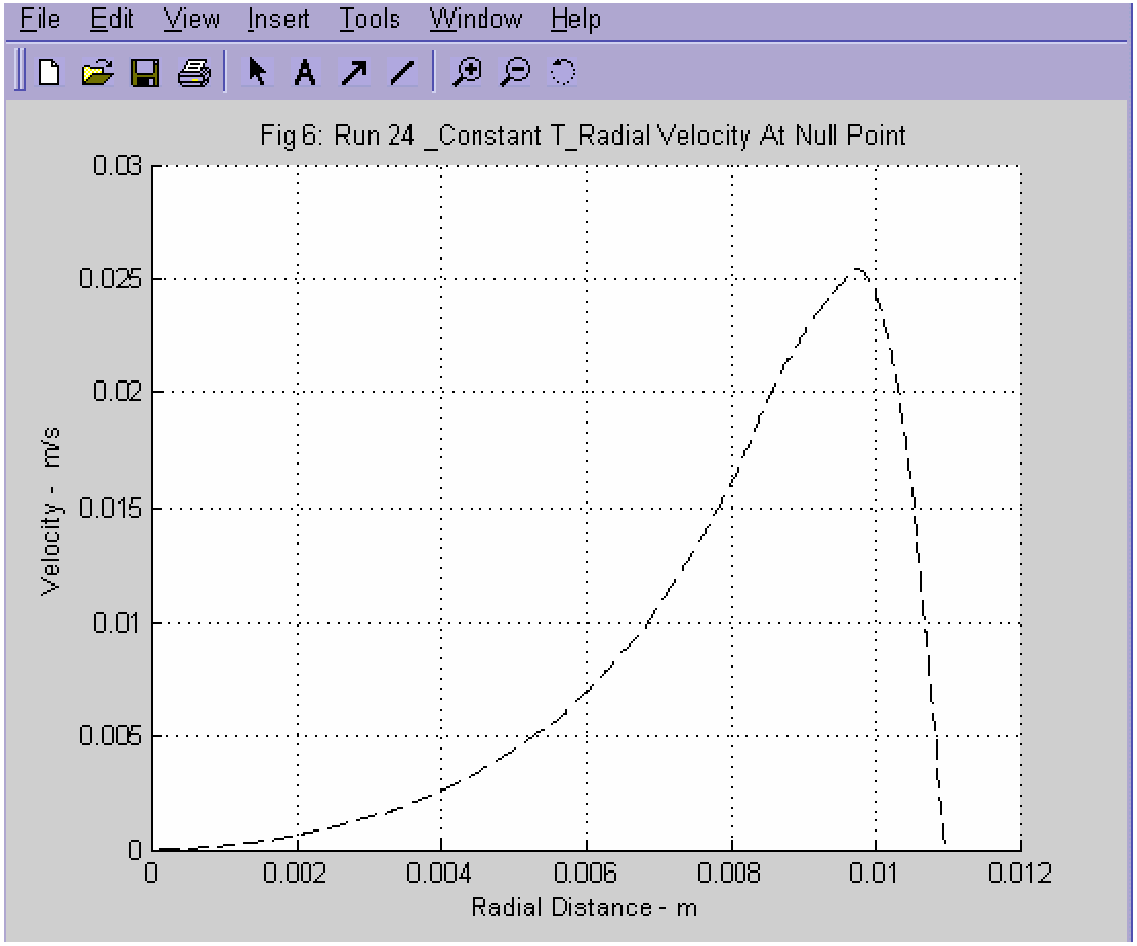
Figure 4A

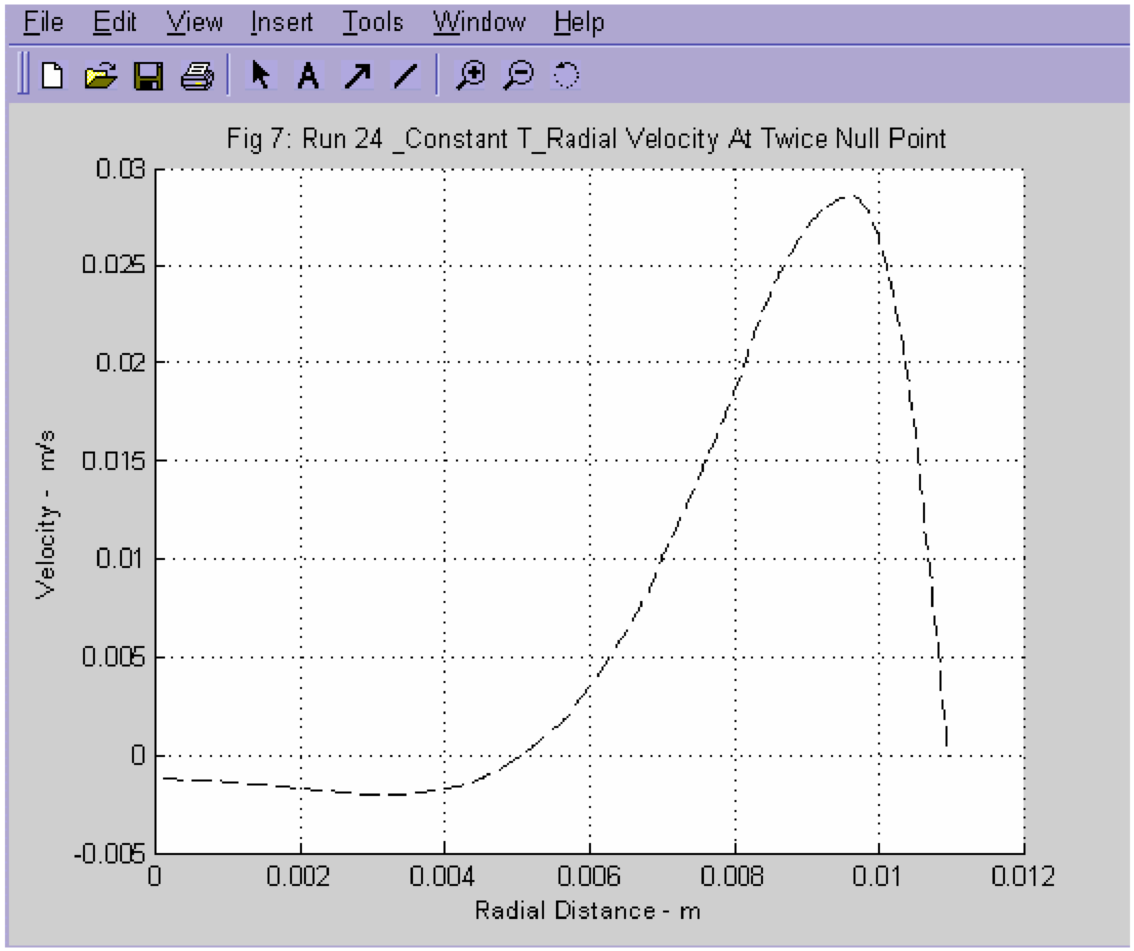
Physical Properties of Water for Experiments - Flux  and Constant Temperature Results

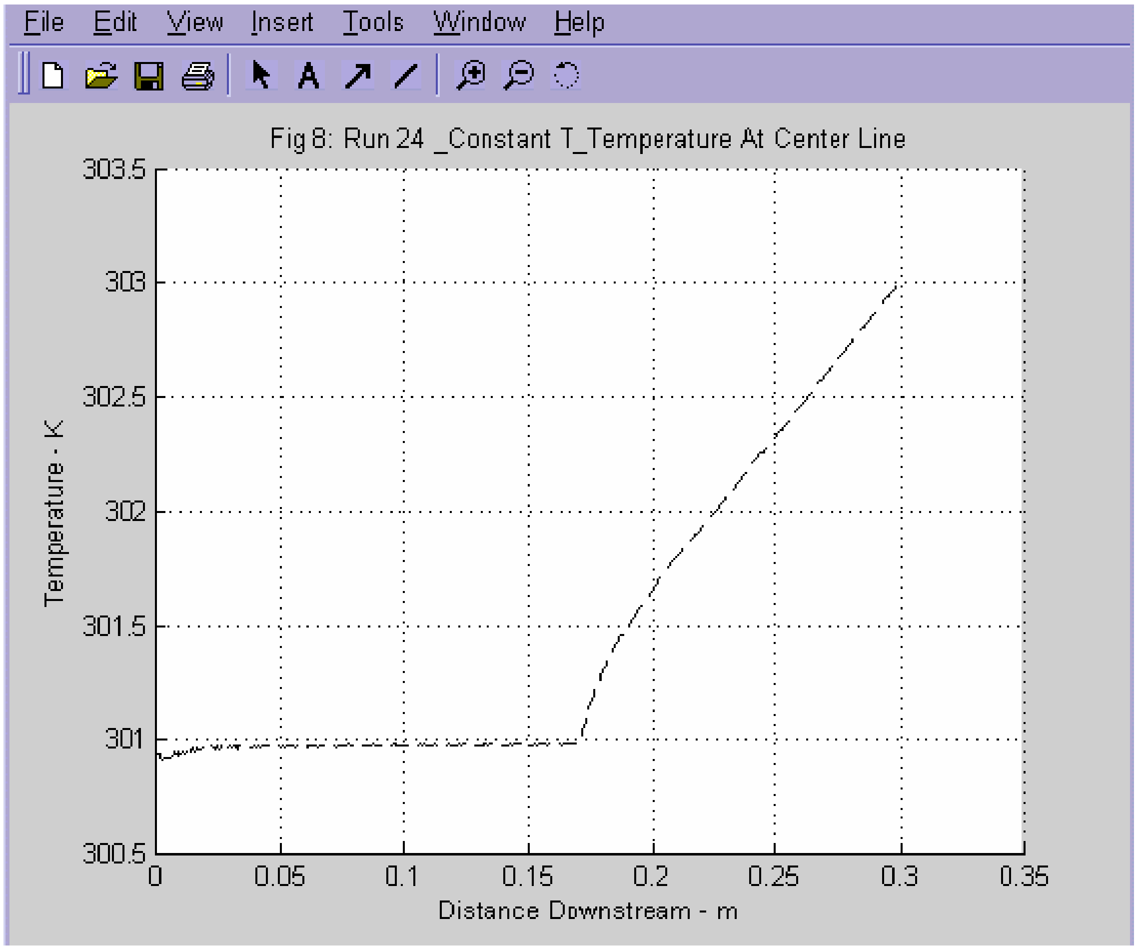
| | Run No ==> | Run 1 | Run 6 | Run 24 | Run 19 | Run 35 | Run 2 | Run_28 | Run 30 | Run 23 | Run 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Inlet, Wall and Reference | | | | | | | | | | | |
| Temperatures | Units | | | | | | | | | | |
| td F (Td) | | 102 | 109.2 | 107.5 | 108.4 | 114.8 | 102 | 117 | 116.8 | 108.3 | 131.5 |
| **td K (Td K)** | Kelvins | 312.0388889 | 316.0388889 | 315.0944444 | 315.5944444 | 319.15 | 312.03889 | 320.372222 | 320.26111 | 315.5388889 | 328.4277778 |
| to F (To) | | 80.2 | 80.9 | 82.1 | 81.2 | 82.2 | 80.3 | 71.4 | 73.4 | 82.3 | 78.5 |
| **to K (To K)** | Kelvins | 299.9277778 | 300.3166667 | 300.9833333 | 300.4833333 | 301.03889 | 299.98333 | 295.038889 | 296.15 | 301.0944444 | 298.9833333 |
| to C (To C) | | 26.77777778 | 27.16666667 | 27.83333333 | 27.33333333 | 27.888889 | 26.833333 | 21.8888889 | 23 | 27.94444444 | 25.83333333 |
| | | | | | | | | | | | |
| Density | | | | | | | | | | | |
| low T | C | 26.7 | 27.1 | 27.8 | 27.3 | 27.8 | 26.8 | 21.8 | 23 | 27.9 | 25.8 |
| low rho | gms/milliter | 0.9966243 | 0.9965146 | 0.996319 | 0.9964591 | 0.996319 | 0.996597 | 0.9988444 | 0.9975674 | 0.9962907 | 0.9968657 |
| high T | C | 26.8 | 27.2 | 27.9 | 27.4 | 27.9 | 26.9 | 21.9 | 23.1 | 28 | 25.9 |
| high rho | gms/milliter | 0.996597 | 0.9964869 | 0.9962907 | 0.9964313 | 0.9962907 | 0.9965696 | 0.9988219 | 0.9975437 | 0.9962623 | 0.9968393 |
| rho | grams/milliter | 0.996603067 | 0.996496133 | 0.996309567 | 0.996449833 | 0.9962938 | 0.9965879 | 0.9988244 | 0.9975674 | 0.996278078 | 0.9968569 |
| **rho0** | **kg/m^3** | **996.6030667** | **996.4961333** | **996.3095667** | **996.4498333** | **996.29384** | **996.58787** | **998.8244** | **997.5674** | **996.2780778** | **996.8569** |
| | | | | | | | | | | | |
| Dynamic Viscosity | | | | | | | | | | | |
| low T | C | 26 | 27 | 27 | 27 | 27 | 26 | 21 | 23 | 27 | 25 |
| low mu | centipoise | 0.8737 | 0.8545 | 0.8545 | 0.8545 | 0.8545 | 0.8737 | 0.981 | 0.9358 | 0.8545 | 0.8937 |
| high T | C | 27 | 28 | 28 | 28 | 28 | 27 | 22 | 24 | 28 | 26 |
| hgh Mu | centipoise | 0.8545 | 0.836 | 0.836 | 0.836 | 0.836 | 0.8545 | 0.9579 | 0.9142 | 0.836 | 0.8737 |
| mu | centipoise | 0.858766667 | 0.851416667 | 0.839083333 | 0.848333333 | 0.8380556 | 0.8577 | 0.96046667 | 0.9358 | 0.837027778 | 0.877033333 |
| **muo** | **kg/m-s** | **0.000858767** | **0.000851417** | **0.000839083** | **0.000848333** | **0.0008381** | **0.0008577** | **0.00096047** | **0.0009358** | **0.000837028** | **0.000877033** |
| | | | | | | | | | | | |
| Thermal Conductivity | | | | | | | | | | | |
| low T | C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| low k | BTU/hr-ft-F | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 | 0.343 |
| high T | C | 37.77777778 | 37.77777778 | 37.77777778 | 37.77777778 | 37.777778 | 37.777778 | 37.7777778 | 37.777778 | 37.77777778 | 37.77777778 |
| high k | BTU/hr-ft-F | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 | 0.363 |
| k | BTU/hr-ft-F | 0.357176471 | 0.357382353 | 0.357735294 | 0.357470588 | 0.3577647 | 0.3572059 | 0.35458824 | 0.3551765 | 0.357794118 | 0.356676471 |
| **ko** | **J/s-m-K** | **0.618022447** | **0.618378685** | **0.618989379** | **0.618531359** | **0.6190403** | **0.6180733** | **0.61354402** | **0.6145618** | **0.619091162** | **0.617157297** |
| | | | | | | | | | | | |
| Heat Capacity | | | | | | | | | | | |
| low T | C | 26 | 27 | 27 | 27 | 27 | 26 | 21 | 23 | 27 | 25 |
| low Cp | cal/gm-C | 0.99885 | 0.99878 | 0.99878 | 0.99878 | 0.99878 | 0.99885 | 0.99933 | 0.99912 | 0.99878 | 0.99892 |
| high T | C | 27 | 28 | 28 | 28 | 28 | 27 | 22 | 24 | 28 | 26 |
| high Cp | cal/gm-C | 0.99878 | 0.99873 | 0.99873 | 0.99873 | 0.99873 | 0.99878 | 0.99921 | 0.99902 | 0.99873 | 0.99885 |
| Cp | cal/gm-C | 0.998795556 | 0.998771667 | 0.998738333 | 0.998763333 | 0.9987356 | 0.9987917 | 0.99922333 | 0.99912 | 0.998732778 | 0.998861667 |
| **Cpo** | **J/kg-K** | **4178.960604** | **4178.860653** | **4178.721187** | **4178.825787** | **4178.7096** | **4178.9443** | **4180.75043** | **4180.3181** | **4178.697942** | **4179.237213** |

| **Figure 4B** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Run No ==> | Run 1 | Run 6 | Run 24 | Run 19 | Run 35 | Run 2 | Run_28 | Run 30 | Run 23 | Run 3 |
| Coefficient of Expansion | | | | | | | | | | | |
| low T | C | 26 | 27 | 27 | 27 | 27 | 26 | 21 | 23 | 27 | 25 |
| low beta | 1/F | 0.14822 | 0.15343 | 0.15343 | 0.15343 | 0.15343 | 0.14822 | 0.12054 | 0.13194 | 0.15343 | 0.14289 |
| high T | C | 27 | 28 | 28 | 28 | 28 | 27 | 22 | 24 | 28 | 26 |
| high beta | 1/F | 0.15343 | 0.15859 | 0.15859 | 0.15859 | 0.15859 | 0.15343 | 0.12628 | 0.13746 | 0.15859 | 0.14822 |
| beta | 1/F | 0.152272222 | 0.15429 | 0.15773 | 0.15515 | 0.1580167 | 0.1525617 | 0.12564222 | 0.13194 | 0.158303333 | 0.147331667 |
| **betao (alpha)** | **1/Kelvins** | **0.00027409** | **0.000277722** | **0.000283914** | **0.00027927** | **0.0002844** | **0.0002746** | **0.00022616** | **0.0002375** | **0.000284946** | **0.000265197** |
| | | | | | | | | | | | |
| Velocity | | | | | | | | | | | |
| **a** | **m** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** | **0.01095** |
| gms/sec | gm/s | 5.18 | 1.97 | 4.74 | 7.31 | 4.21 | 9.82 | 12.2 | 10.2 | 3.88 | 10.95 |
| v(average) | m/s | 0.013798434 | 0.00524823 | 0.012630086 | 0.019475302 | 0.011218 | 0.0261588 | 0.03242597 | 0.0271444 | 0.010338878 | 0.029161077 |
| Uin -v(maximum) | m/s | 0.027596868 | 0.01049646 | 0.025260171 | 0.038950603 | 0.0224361 | 0.0523176 | 0.06485194 | 0.0542888 | 0.020677756 | 0.058322153 |
| | | | | | | | | | | | |
| Temp Coeff of Viscosity | | | | | | | | | | | |
| low Temperature | C | 26 | 27 | 27 | 27 | 27 | 26 | 21 | 23 | 27 | 25 |
| low b | 1/F | 0.012418 | 0.012285 | 0.012285 | 0.012285 | 0.012285 | 0.012418 | 0.013302 | 0.012946 | 0.012285 | 0.012616 |
| high Temperature | C | 27 | 28 | 28 | 28 | 28 | 27 | 22 | 24 | 28 | 26 |
| high b | 1/F | 0.012285 | 0.012116 | 0.012116 | 0.012116 | 0.012116 | 0.012285 | 0.013139 | 0.012781 | 0.012116 | 0.012418 |
| **b** | **1/Kelvins** | 0.0221662 | 0.0220623 | 0.0218595 | 0.0220116 | 0.0218426 | 0.0221529 | 0.0236828 | 0.0233028 | 0.0218257 | 0.0224118 |
| | | | | | | | | | | | |
| Parameters | | | | | | | | | | | |
| Re | Reynolds Number | 175.3440187 | 67.26055612 | 164.213789 | 250.48817 | 146.0312 | 332.82233 | 369.244392 | 316.84985 | 134.7498312 | 362.9396318 |
| Gr | Grashof Number | 57581.45373 | 77037.56027 | 72750.80382 | 74991.12394 | 93769.4 | 57567.437 | 79804.1559 | 83810.029 | 75102.80873 | 129932.195 |
| Gr/Re | | 328.3913199 | 1145.360144 | 443.0249389 | 299.3799026 | 642.11893 | 172.96747 | 216.128281 | 264.51024 | 557.3499282 | 357.9994677 |
| ln (Gr/Re) | | 5.794205946 | 7.043474403 | 6.093626064 | 5.701713344 | 6.4647735 | 5.1531036 | 5.37587213 | 5.5778799 | 6.32319328 | 5.880531499 |
| Pr | | 5.806831266 | 5.753677629 | 5.664548406 | 5.731378302 | 5.6571292 | 5.7991185 | 6.54471606 | 6.3654157 | 5.649711171 | 5.939053725 |
| Psi | | 0.268457311 | 0.346868383 | 0.308461833 | 0.332619733 | 0.3955938 | 0.2670655 | 0.59996427 | 0.5618564 | 0.315260111 | 0.659903 |
| | | | | | | | | | | | |
| ***Femlab - Boundary Flux*** | | | | | | | | | | | |
| z | m | 0.285 | 0.061 | 0.22 | 0.43 | 0.16 | 0.931 | 0.87 | 0.624 | 0.161 | 0.573 |
| Z | | 26.02739726 | 5.570776256 | 20.0913242 | 39.26940639 | 14.611872 | 85.022831 | 79.4520548 | 56.986301 | 14.70319635 | 52.32876712 |
| Z* | | 0.008136752 | 0.004582054 | 0.006875181 | 0.008706799 | 0.0056301 | 0.0140221 | 0.01046528 | 0.0089937 | 0.006147631 | 0.007727506 |
| Gz | | 122.8991637 | 218.2427217 | 145.4507143 | 114.8527675 | 177.61735 | 71.316231 | 95.5540482 | 111.18836 | 162.6642911 | 129.4078588 |
| Z**1000 | | **8.136751873** | **4.582054293** | **6.875181086** | **8.706799334** | **5.6300805** | **14.022053** | **10.4652814** | **8.9937474** | **6.147630764** | **7.72750596** |
| | | | | | | | | | | | |
| ***Femlab - Constant Temp*** | | | | | | | | | | | |
| z | m | 0.2185 | 0.052 | 0.171 | 0.324 | 0.1265 | 0.668 | 0.62 | 0.455 | 0.122 | 0.41 |
| Z | | 19.9543379 | 4.748858447 | 15.61643836 | 29.5890411 | 11.552511 | 61.004566 | 56.6210046 | 41.552511 | 11.14155251 | 37.44292237 |
| Z* | | 0.006238176 | 0.003906013 | 0.005343891 | 0.006560472 | 0.0044513 | 0.0100609 | 0.00745802 | 0.0065579 | 0.004658453 | 0.00552928 |
| Gz | | 160.3032569 | 256.0155005 | 187.129574 | 152.4280557 | 224.65436 | 99.394328 | 134.083906 | 152.48689 | 214.6635317 | 180.8553734 |
| Z**1000 | | **6.238176436** | **3.906013496** | **5.343890753** | **6.560472057** | **4.4512824** | **10.060936** | **7.45801661** | **6.5579408** | **4.658453125** | **5.529280007** |


File Edit View Insert Tools Window Help
Fig 5: Run 24 _Constant T_Center Line Velocity
Velocity - m/s
Distance Downstream m
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35


File Edit View Insert Tools Window Help
Fig 6: Run 24 _Constant T_Radial Velocity At Null Point
Velocity - m/s
0
0.005
0.01
0.015
0.02
0.025
0.03
0
0.002
0.004
0.006
0.008
0.01
0.012
Radial Distance - m


File Edit View Insert Tools Window Help
Fig 7: Run 24 _Constant T_Radial Velocity At Twice Null Point
Velocity - m/s
Radial Distance - m
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
0
0.002
0.004
0.006
0.008
0.01
0.012


File Edit View Insert Tools Window Help
Fig 8: Run 24 _Constant T_Temperature At Center Line
Temperature - K
303.5
303
302.5
302
301.5
301
300.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Distance Downstream - m

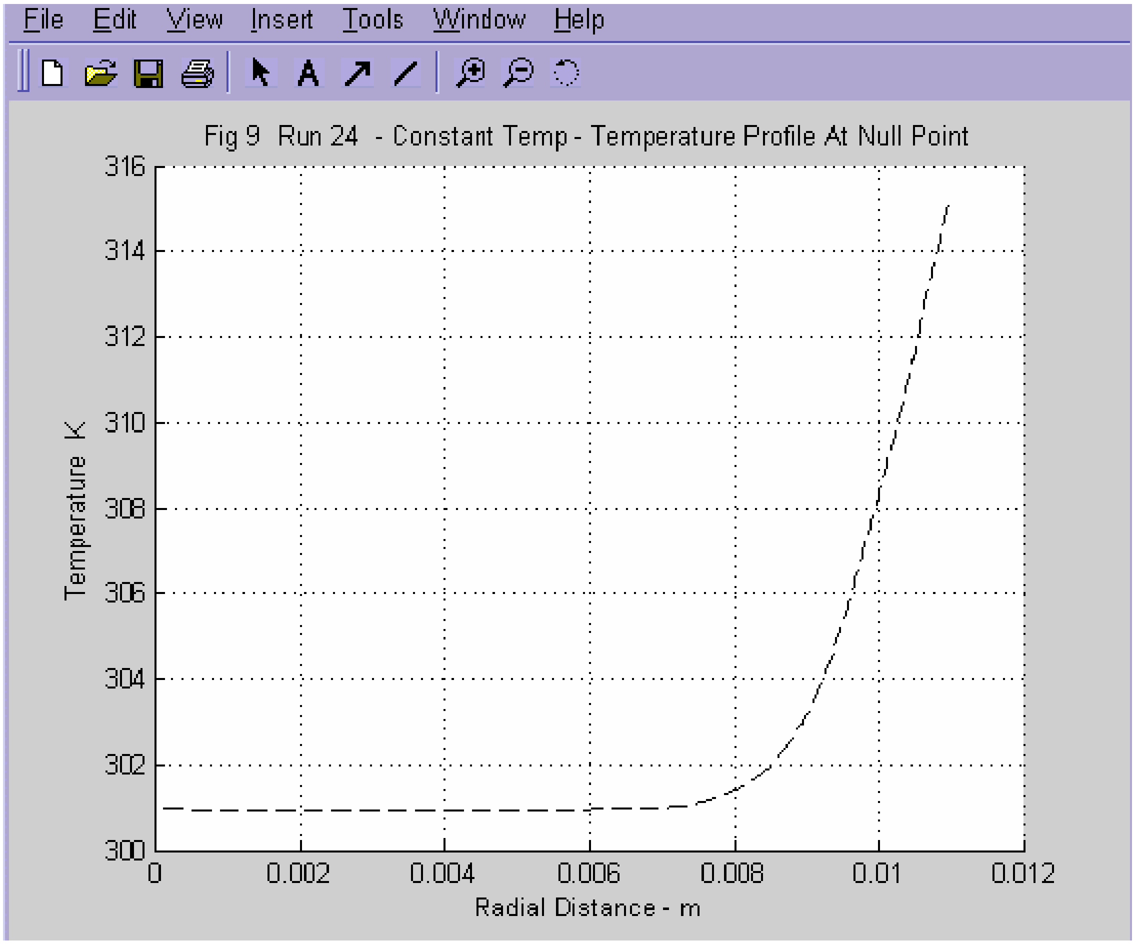

File Edit View Insert Tools Window Help
Fig 9 Run 24 - Constant Temp - Temperature Profile At Null Point
Temperature K
316
314
312
310
308
306
304
302
300
0
0.002
0.004
0.006
0.008
0.01
0.012
Radial Distance - m

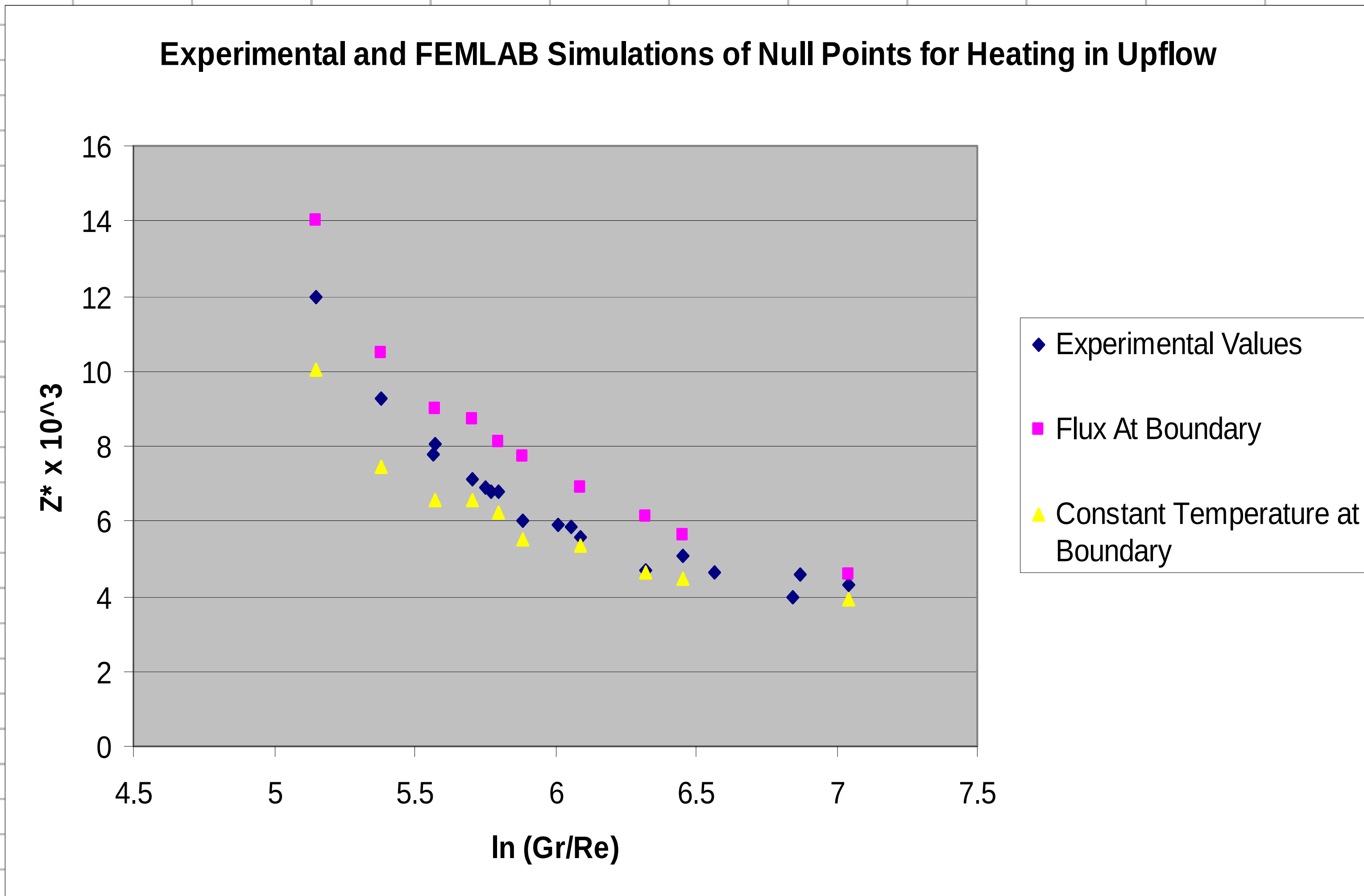
Figure 10A Results of Study - Experimental Data From Page 243 Thesis

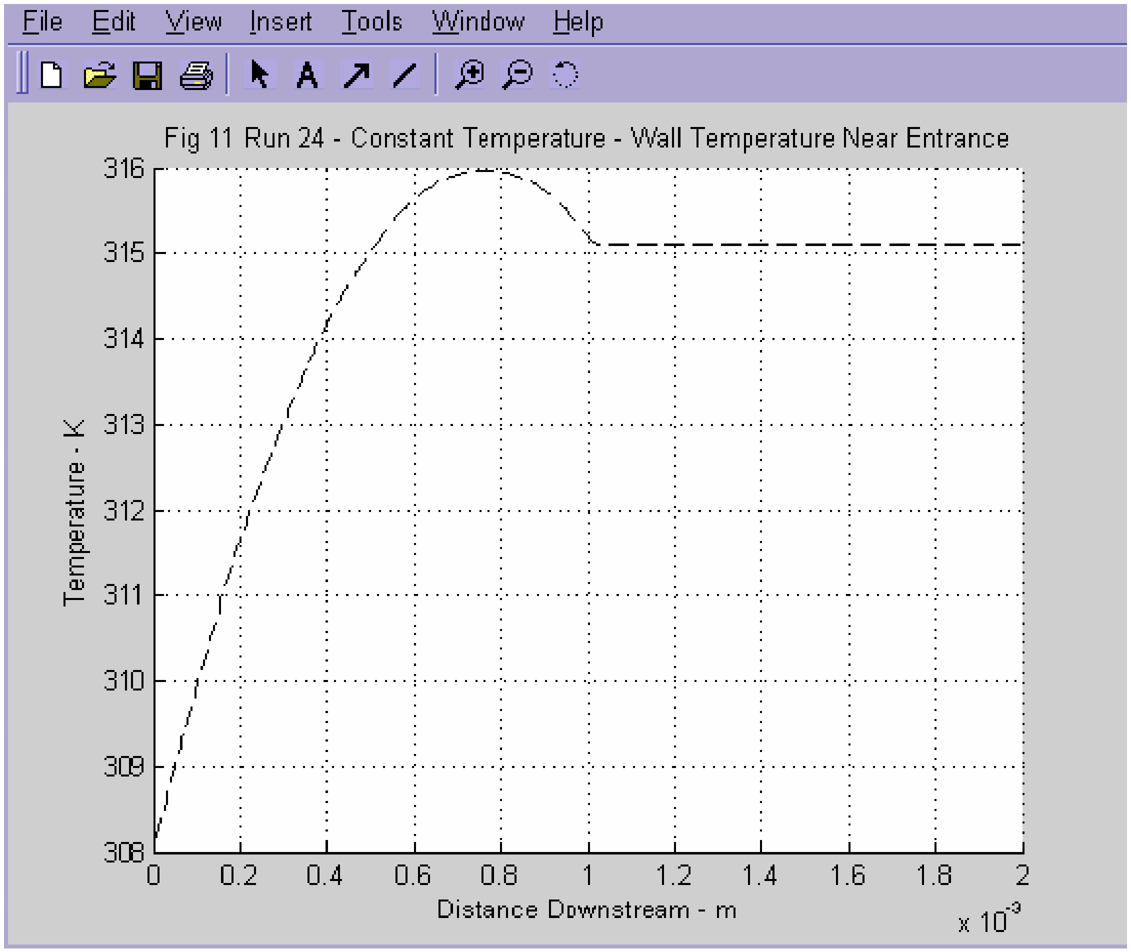
| | | | | | | Experimental----------> | | Femlab-Flux-------------> | | Femlab-Temp |
|---|---|---|---|---|---|---|---|---|---|---|
| Run | Pr | Gr | Re | Gznull | Gr/Re | ln(Gr/Re) | Z* x 10^3 | ln(Ge/Re) | Z* x 10^3 | Z* x 10^3 |
| 1 | 5.79 | 57500 | 175 | 147 | 328.5714286 | 5.794754253 | 6.8027211 | 5.79675 | 8.136 | 6.238 |
| 2 | 5.78 | 57500 | 333 | 83.7 | 172.6726727 | 5.151397737 | 11.947431 | 5.151 | 14.02 | 10.06 |
| 3 | 5.93 | 130000 | 362 | 166 | 359.1160221 | 5.883645518 | 6.0240964 | 5.8805 | 7.72 | 5.528 |
| 6 | 5.74 | 76900 | 67.3 | 233 | 1142.644874 | 7.041100919 | 4.2918455 | 7.043 | 4.582 | 3.906 |
| 15 | 5.85 | 66800 | 156 | 171 | 428.2051282 | 6.059602352 | 5.8479532 | | | |
| 16 | 5.81 | 66500 | 254 | 129 | 261.8110236 | 5.56762296 | 7.751938 | | | |
| 19 | 5.72 | 75000 | 250 | 140 | 300 | 5.703782475 | 7.1428571 | 5.7 | 8.706 | 6.56 |
| 20 | 5.71 | 74000 | 231 | 147 | 320.3463203 | 5.769402662 | 6.8027211 | | | |
| 21 | 5.69 | 75600 | 186 | 170 | 406.4516129 | 6.007464888 | 5.8823529 | | | |
| 23 | 5.66 | 74400 | 134 | 214 | 555.2238806 | 6.319371421 | 4.6728972 | 6.23 | 6.147 | 4.658 |
| 24 | 5.67 | 72500 | 164 | 180 | 442.0731707 | 6.091475413 | 5.5555556 | 6.093 | 6.875 | 5.34 |
| 25 | 5.9 | 76100 | 242 | 145 | 314.4628099 | 5.750865818 | 6.8965517 | | | |
| 26 | 5.88 | 77000 | 108 | 216 | 712.962963 | 6.569429474 | 4.6296296 | | | |
| 27 | 5.87 | 76400 | 81.2 | 251 | 940.8866995 | 6.846822728 | 3.9840637 | | | |
| 28 | 6.53 | 79900 | 368 | 108 | 217.1195652 | 5.380448194 | 9.2592593 | 5.375 | 10.46 | 7.458 |
| 30 | 6.38 | 83600 | 317 | 124 | 263.7223975 | 5.574897025 | 8.0645161 | 5.5748 | 8.99 | 6.55 |
| 34 | 5.69 | 95900 | 99.8 | 219 | 960.9218437 | 6.867893078 | 4.56621 | | | |
| 35 | 5.66 | 93000 | 146 | 198 | 636.9863014 | 6.45674815 | 5.0505051 | 6.465 | 5.63 | 4.451 |

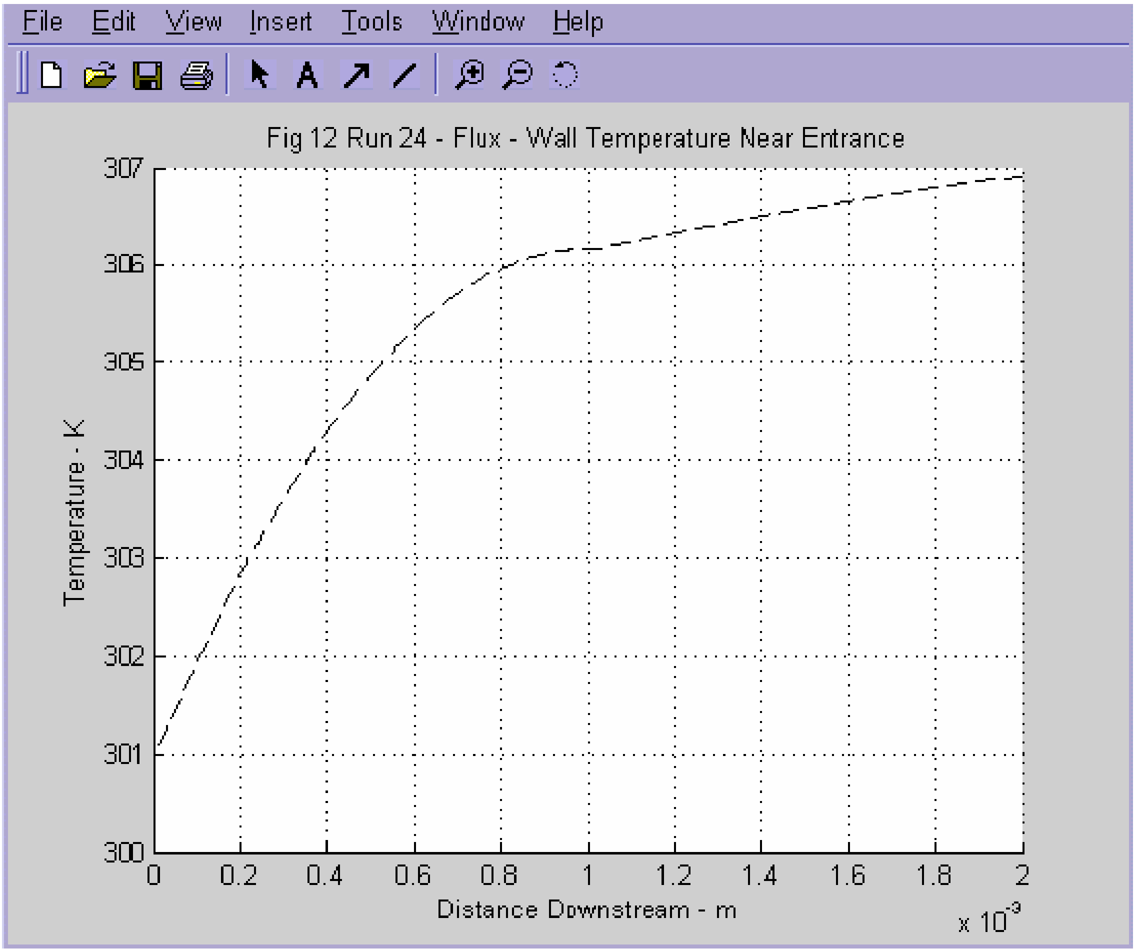
<-------------------------- Plot Data------------------------------------------------->

| ln(Gr/Re) | Experimental | Z* x 10^3 Flux | Z* x 10^3 Temp |
|---|---|---|---|
| 5.79475425 | 6.802721088 | 8.136 | 6.238 |
| 5.15139774 | 11.9474313 | 14.02 | 10.06 |
| 5.88364552 | 6.024096386 | 7.72 | 5.528 |
| 7.04110092 | 4.291845494 | 4.58 | 3.906 |
| 6.05960235 | 5.847953216 | | |
| 5.56762296 | 7.751937984 | | |
| 5.70378247 | 7.142857143 | 8.706 | 6.56 |
| 5.76940266 | 6.802721088 | | |
| 6.00746489 | 5.882352941 | | |
| 6.31937142 | 4.672897196 | 6.147 | 4.658 |
| 6.09147541 | 5.555555556 | 6.875 | 5.34 |
| 5.75086582 | 6.896551724 | | |
| 6.56942947 | 4.62962963 | | |
| 6.84682273 | 3.984063745 | | |
| 5.38044819 | 9.259259259 | 10.46 | 7.458 |
| 5.57489703 | 8.064516129 | 8.99 | 6.55 |
| 6.86789308 | 4.566210046 | | |
| 6.45674815 | 5.050505051 | 5.63 | 4.451 |

Figure 10 B Plot of Results




File Edit View Insert Tools Window Help
Fig 11 Run 24 - Constant Temperature - Wall Temperature Near Entrance
316
315
314
313
312
311
310
309
308
Temperature - K
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Distance Downstream - m
x 10-3


File Edit View Insert Tools Window Help
Fig 12 Run 24 - Flux - Wall Temperature Near Entrance
Temperature - K
307
306
305
304
303
302
301
300
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Distance Downstream - m
x 10-3

# A VIRTUAL UNIT OPERATIONS LABORATORY

PATRICK J. FLEMING, MICHAEL E. PAULAITIS
*Johns Hopkins University • Baltimore, MD 21218*

Chemical engineering students typically take a unit operations laboratory course and a capstone design course in their senior year. Experiments in the unit operations laboratory might focus on a practical design problem, while the design course is often based on a case study drawn from the experience of a practicing engineer. These courses are intended to confront students with problems that are not well-defined; problems that will be encountered in the complex "real world" of chemical engineering practice. The capstone design course, in particular, requires integration of basic science and engineering facts, theory, and quantitative problem-solving skills learned in the earlier years of the curriculum. The scope of these "real world" problems is, in most cases, limited by the traditional classroom or laboratory settings in which they are taught. Moreover, laboratory experiences that complement classroom instruction in other courses are the exception rather than the rule. A major constraint to developing new, innovative teaching laboratories that offer more practical engineering experiences to undergraduates is that operating and maintaining them is expensive in terms of manpower, space, and equipment.

In this paper, we describe a virtual laboratory now being created in the Department of Chemical Engineering at Johns Hopkins that features real-time dynamic simulations of each experiment in our undergraduate unit operations laboratory course. The overall goal of this virtual laboratory project is to enable our students to experience a broad range of design, scale-up (and even start-up) problems, normal and unusual operating conditions, and safety issues going well beyond the scope of experiences that could ever be contemplated in a traditional chemical engineering unit operations laboratory.

***Michael Paulaitis*** *is Chair of the Department of Chemical Engineering at Johns Hopkins University. He holds degrees from Princeton University (BS, 1968), Stanford University (MS, 1970), and the University of Illinois (PhD, 1976). His research interests include chemical engineering thermodynamics, statistical/molecular thermodynamics, and molecular simulations.*
***Patrick Fleming*** *is Principal Research Scientist in the Department of Chemical Engineering at Johns Hopkins University. His BA (1969) is from Kalamazoo College and his PhD (1975) is from the University of Michigan. His research interests include simulation of protein interactions and solvation and protein folding.*



## THE VIRTUAL LABORATORY

We have so far built, parameterized, and in-house tested a batch distillation column simulation model as the first experiment in the virtual laboratory. The simulation and process control software, provided by GSE Systems, Inc.,[1] is that used in the chemical, pharmaceutical, energy, process, and manufacturing industries (see Figure 1).

The software comprises a D/3 distributed control system (D/3 DCS™) with SimSuite Pro™ simulation software and runs on PC-level computers. The D/3 distributed control software permits display and control of any process communicating via analog or digital signal. SimSuite Pro is stand-alone software that is used to build, test, and run process simulation models dynamically and in real time. Additional Active X™-based software donated by GSE enables display of variable output, control panel, and process graphics inside a web browser. With these software modules we are able to control the pilot-scale distillation column now used in our chemical engineering unit operations laboratory, collect data from the column during an experiment, or simulate it in real time.

The actual column consists of six identical bubble cap trays, each fitted with a liquid sampling port and a copper-constantan thermocouple to measure the temperature of the entering vapor. The reboiler is a 20-liter vessel with variable electric heating. The total condenser at the top of the column is supplied with cooling water at 25°C and 30 psig. Reflux to the column is controlled by an electrically operated solenoid valve. During normal operations, this valve is in the "open" position when energized, and the liquid condensate is collected as product. In the event of electrical failure, gravity causes the valve to close such that all liquid condensate is returned to the column. The reflux ratio is controlled by a timer on the solenoid that periodically opens and closes the valve.

The distillation column simulation model is a computer program with graphic displays on an instructor's workstation and on an operator's console. It performs stagewise vapor-liquid equilibrium calculations based on tray pressure, mixture composition and enthalpy, and vapor-liquid traffic in the column. The model considers tray holdup, external heat transfer, and column geometry. It calculates in real time the temperature, vapor and liquid flow rates, enthalpies, and compositions for each tray. During equilibrium staged operations, the column is modeled by solving simultaneously component material and energy balances, and vapor-liquid equilibrium relationships using Wilson equation activity coefficients to account for thermodynamic non-idealities in the liquid phase.

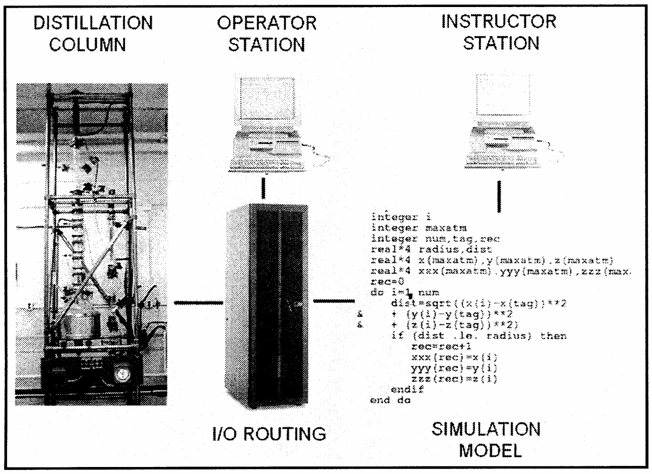



***Figure 1.*** *Configuration of process control and simulation modules for the distillation column experiment. The operator's station communicates with the simulator by means of a process control module that translates input/output signals to the appropriate format and display. The actual distillation column (or any other unit operations equipment) may also be controlled from the same operator's station by using a similar active display that is connected to the input/output addresses of components on the real equipment. The instructor's station may also control the simulation model or initiate malfunctions during the experiment.*

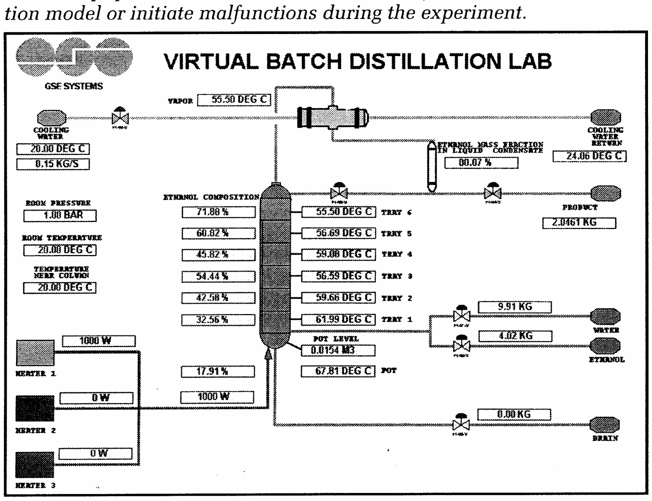



***Figure 2.*** *Active computer display seen by students during simulated operation of the distillation column.*

The simulation model was validated using temperature and composition data collected at each tray for ethanol-water mixtures in the actual distillation column. During transient regimes, it is modeled by allowing the stage efficiencies, Wilson equation parameters, and column heat losses to be functions of time. These functions were determined by tuning the model using data collected for ethanol-water mixtures in the actual column during transient operations. All the measurements were made by a team of Hopkins undergraduate chemical engineers as a summer project.

Flooding in the column, loss of cooling water, failed valves, or other malfunctions can be simulated from the instructor's station of the virtual laboratory. During development, the station is used to tune the running model, and during a virtual distillation experiment, it is used to set initial conditions, control model execution, and insert interventions, such as malfunctions. The system is capable of maintaining up to 100 different initial conditions: values for all process variables, remote functions, and malfunctions.

The simulation model exists not only as a computer program running in the background but is also experienced by the students as an on-screen graphic image depicting a distillation apparatus. Stream flows, temperatures, liquid compositions, and volumes are displayed in real time (see Figure 2). Operation of the distillation process is controlled by students from the screen by mouse-clicking on the various components. Students enter their choice of operating parameters such as heating current or valve openings or closings. The same graphic image may be used to monitor and control the actual distillation column. In principal, the student need not know if the virtual or actual distillation process is connected to the display, although in practice the virtual process provides more on-screen information such as tray compositions.

Data on many process variables is continuously collected and may be viewed in real time (see Figure 3) or downloaded to a spreadsheet program for later analysis.

## CURRICULUM IMPLEMENTATION

The distillation simulation model was used for the first time during the 2000/01 academic year in two chemical engineering courses: Chemical Engineering Process Design and Separations Processes. The latter is a junior-year chemical engineering core course taught in the Spring semester. Chemical Engineering Process Design is taken by seniors in the same semester. In the previous (Fall) semester the seniors normally have taken Unit Operations laboratory. Batch distillation is a required experiment in this laboratory course, and the students are required to design and then carry out experiments to characterize the actual batch distillation column.

The context and motivation for the experiment is to train personnel with limited technical backgrounds to separate several liquid solvent mixtures using the actual pilot-scale batch distillation column to accomplish the task. The first objective is to decide what data needs to be collected to accurately characterize the column. As a team, the students identify tray efficiencies as the key information they need and carry out McCabe-Thiele analyses to get this information from the measured tray temperatures and compositions. The students are also required to select a suitable "model" system (ethanol/water is selected in the end), specify the operating conditions for the distillation of this mixture, determine startup and shutdown procedures, etc. They also consider safety, cost, and environmental impact.

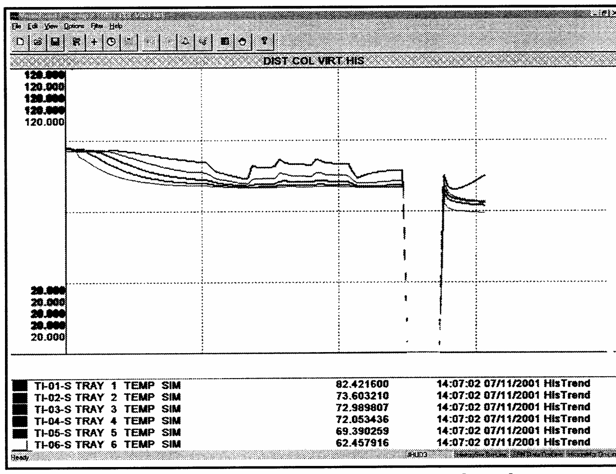



***Figure 3.*** *Real-time display of tray temperature data during simulated operation of the distillation column. Note the abrupt changes in temperature in all trays signifying a recharging of the reboiler.*

Once the actual distillation column has been characterized, the students move on to another experiment in the unit operations laboratory. The stated overall objective to train unskilled personnel to use the equipment to separate other solvent mixtures is never met or even considered beyond motivating the original experiments. With the ability to carry out essentially an unlimited variety of separations using the virtual distillation column, it is now straightforward to ask the students to run the virtual column to complete the overall objective of the experiment.

We implemented this "mini-design" project in Chemical Engineering Process Design using a problem-based learning approach.[2] Teams of senior chemical engineers who were now familiar with operating the actual distillation column were given the assignment of repeating the experiment using the simulation model. Each team of three seniors was introduced to the distillation simulation model during an initial two-hour session in which they operated the simulation according to prescribed instructions contained in a tutorial developed for the mini-design project of one of the authors (PJF).

During the second session, the students operated the virtual column independently. They charged the column pot with an ethanol/water mixture, brought the column to equilibration, and gathered data on the temperature and composition of each tray. For the third session, each team was given a virtual column with a different binary mixture, listed in Table 1. Their task was to determine the optimal operating parameters for performing a single batch distillation enrichment of the mixture and to prepare a training program and manual to teach inexperienced operators how to carry out the specific separation. Although guidelines on size and format were provided, each team could decide what the best content of such a training manual would be.

The juniors in the Separations Processes course served as the untrained personnel in this project. The timing of their involvement was coordinated such that they were just completing course work on the fundamentals of distillation. Teams of three juniors were paired with the senior teams for this training session. After a short oral presentation by the seniors, the junior operators carried out the batch distillation of their particular binary mixture according to the training manual written by the senior team. After the juniors became familiar with the column operation, the seniors were encouraged to introduce various malfunctions and the juniors were prompted to analyze the problem from displayed process variables and perform corrective actions in real time. One typical malfunction was a failure of the cooling water flow. This failure could be announced by an alarm (if desired by the instructor) or observed on the graphic display as a change in the flow rate of cooling water or an increase in the temperature of the cooling water return. The students usually come to the conclusion that the best corrective procedure is to turn off the heating coils immediately. Malfunctions were also introduced inadvertently. For example, one team accidentally opened the drain valve on the reboiler and quickly drained all of a noxious mixture out of the pot! The versatility of the virtual laboratory allowed us to explore the safety ramifications of this virtual accident, thus providing an educational opportunity that never would have occurred in the actual laboratory experiment.

One of the great advantages of the virtual column is the ability to speed up process clock time by as much as a factor of 10. Thus, instead of waiting hours for the column to reach steady state from a cold start, the start-up can be shortened to minutes. This accelerated process time encourages the students to carry out "what if" experiments that could never be contemplated with the actual distillation column.

Each team (seniors and juniors separately) were required to give oral presentations of their results and to prepare written reports. The seniors were present for the junior team presentations to ask questions. In both oral presentations, the students were asked to describe what they learned during the training session from the perspective of either trainers or trainees. The combination of problem solving, group interaction and delegation of teaching/learning responsibility uses many of the elements described by Wankat for efficient effective teaching.[3]

**TABLE 1**
**Binary Mixtures for Distillation Experiments**

| *Mixture* | *Boiling Points (°C)* | *MSDS Remarks* |
|---|---|---|
| Acetone/Water | 56/100 | Extremely Flammable |
| Acetone/Isopropanol | 56/82 | Extremely Flammable |
| Benzene/Ethanol | 80/79 | Carcinogenic, Azeotrope |
| Heptane/Toluene | 98/110 | Extremely Flammable |
| Isopropanol/Water | 80/100 | Flammable, Azeotrope |
| Methanol/Water | 65/100 | Flammable |
| Methylene Chloride /Ethylene Chloride | 40/83 | Flammable, Extremely Toxic |

In their written evaluations of the project, the seniors were unanimous in their opinion that the opportunity to separate different mixtures and to explore various operating conditions in the virtual column, which they could not do in the unit operations laboratory, allowed them to gain a deeper understanding of distillation. The juniors, in their written evaluations, indicated that the project provided them with a perspective of distillation that went well beyond the fundamentals they learned in the classroom.

To evaluate the continuing impact of this junior-level exposure to virtual distillation experiment, we gave a written survey to the juniors after they had taken the senior unit operations laboratory course and performed the actual distillation experiment. This survey was designed in the format of the Student Assessment of Learning Gains instrument available at the National Institute for Science Education.[4] We asked the students to rank a list of classroom aspects on a five-category scale from "No Help" (1) to "Very Much Help" (5). The questions addressed the value of: a) working in groups; b) hands-on control of simulation; c) the written lab manual; d) verbal instruction by seniors; e) final presentation report; f) impact of virtual lab on real lab. The average scores for the first five aspects (a through e) of the virtual lab were between 2.8 and 3.0 (a little to moderate help); the average score for the last question on the impact of the virtual lab was 3.4 (moderate to much help). Verbal comments from the students that correlate with this high level of impact indicated that their virtual experience instilled more confidence in performing the actual experiment.

It is important to point out that computer simulations such as those described here are not meant to supplant existing real laboratory experiments, but rather to complement them. As stated previously,[5] neither the virtual nor the real laboratory format is "better"—rather, they have different roles in the curriculum. We have used the virtual experiment as a tool to integrate efficient, effective teaching elements[3] into our curriculum. In designing the virtual laboratory experiments, we recognize a key responsibility is to ensure that students continue to have actual hands-on experience with an actual process or device.

## FUTURE

A major goal for the coming year is to make the virtual distillation experiment web-accessible and to create and evaluate virtual laboratory experiments in partnership with several other universities. We also plan to introduce the virtual laboratory in a local high school. We intend to design and evaluate several specific experiments and educational modules around the virtual batch distillation column simulation and use them as templates for a much larger effort to develop a variety of virtual experiments maintained by faculty at other universities. Web-accessibility of virtual experiments has the potential to give chemical engineering students at many educational institutions access to specialized virtual experiments that can complement existing laboratory curricula. Our "mini-design" project demonstrated the important element vertical integration to other courses in the undergraduate chemical engineering curriculum.


## ACKNOWLEDGMENTS

The authors gratefully acknowledge support from the Special Grant Program in the Chemical Sciences, the Camille and Henry Dreyfus Foundation, and computer software, hardware, and training from GSE Systems, Inc., Columbia, Maryland 21045. The efforts of undergraduate students Robin Cohen, Elizabeth Chambers, Gabe Farkas, and Austin Lin are also greatly appreciated. We thank Professor Joseph Katz for help with the actual distillation column operation.


## REFERENCES


1. <www.gses.com> GSE Systems, Inc., is a leading global provider of dynamic simulation and process control systems to the chemical, pharmaceutical, energy, process, and manufacturing industries
2. Woods, D.R., *Problem-Based Learning: How to Gain the Most from Problem-Based Learning,* Hamilton, Ontario, Canada (1994)
3. Wankat, P.C., "Effective, Efficient Teaching," *Chem. Eng. Ed.,* **35**(2), p. 92 (2001)
4. <www.wcer.wisc.edu/nise/>
5. White, S.R., and G.M. Bodner, "Evaluation of Computer-Simulation Experiments in a Senior-Level Capstone ChE Course," *Chem. Eng. Ed.,* **35**(1), p. 34 (1999) ❒

# A Web-Based Case Study for Chemical Engineering Capstone Course

Dr. Lisa Bullard
Dr. Steven Peretti
Patty Niehues
Shannon White

North Carolina State University

NC STATE UNIVERSITY

# Objectives

- Why biotechnology
- What is a case study
- Web-site structure and contents
- Problem statement
- Development of solution
- Exemplary results
- Further development

# Why biotechnology?

- Growing job source for engineers
- Non-traditional application of technology
- Novel process challenges
- Unique regulatory environment
- Inherently multidisciplinary
- Availability of local expertise via ISPE

# What constitutes a case study

- Purpose
- Problem statement
- Supporting website
- Support information
  - Tutorials
  - Information pages
  - Simulation
- Exemplary solution

# Instructor Materials

- Problem Statement
  - Protein characteristics
  - Facility layout
  - Equipment list
- Website description
- Implementation advice
- Resource material

# Web-site structure

- Levels of Access
  - Student Access
    - Problem Statement
    - Deliverables
    - Resources
  - Faculty Access
    - Enhanced Problem Information
    - Password Protected Solution
    - Tips
  - Administrator access
    - Full Read and Write Privileges

# Website
## Student Access

**Case Study Introduction**

**Problem Statement**

**Deliverables**

- Executive Summary
- Introduction
- Product Line Determination/ Economic Analysis
- Process Description
- Process Controls
- Regulatory Requirements

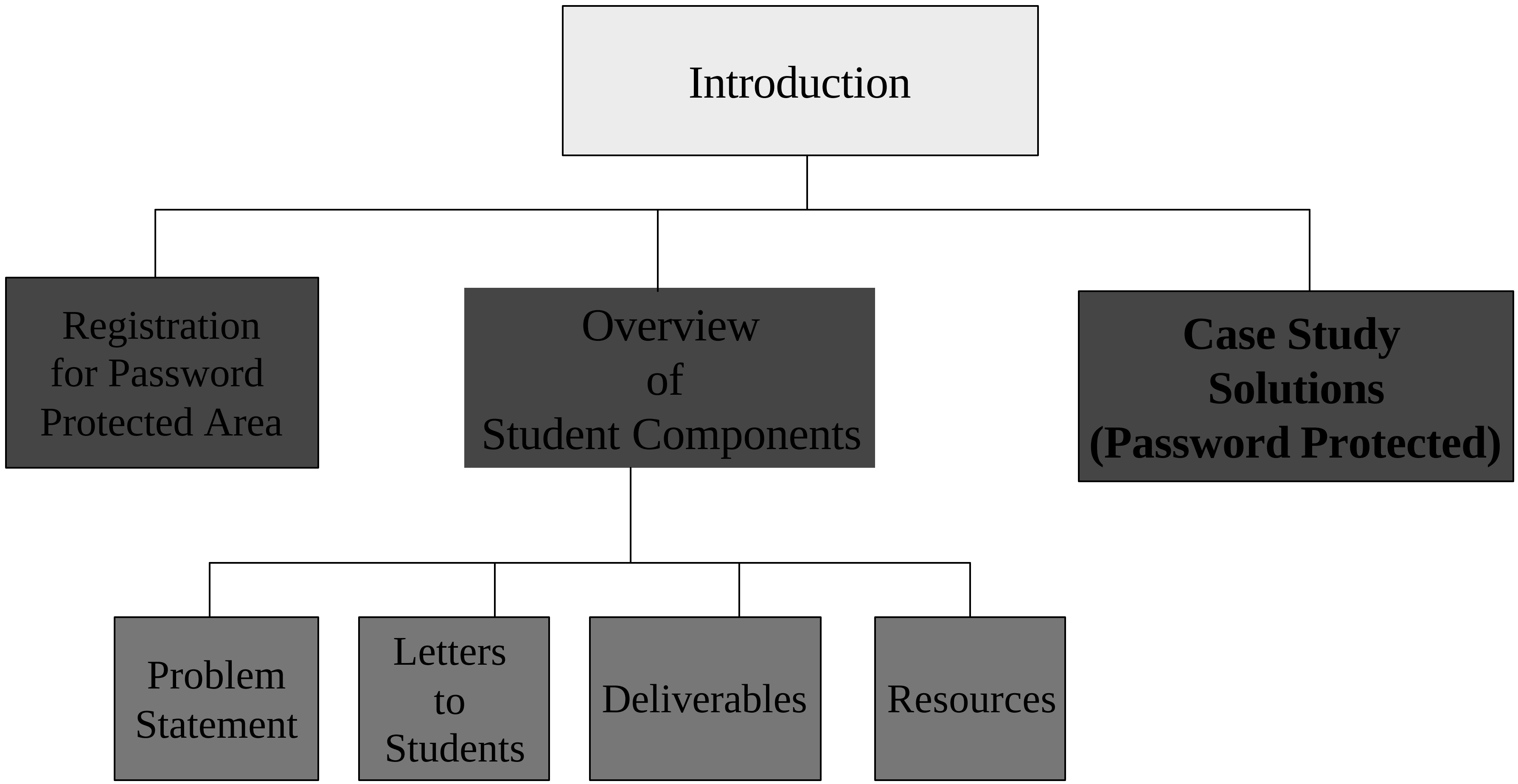
- Validation
- Waste Management
- Facility Design
- Product Line Costs
- Conclusions

**Letters from Students**

**Tutorials**

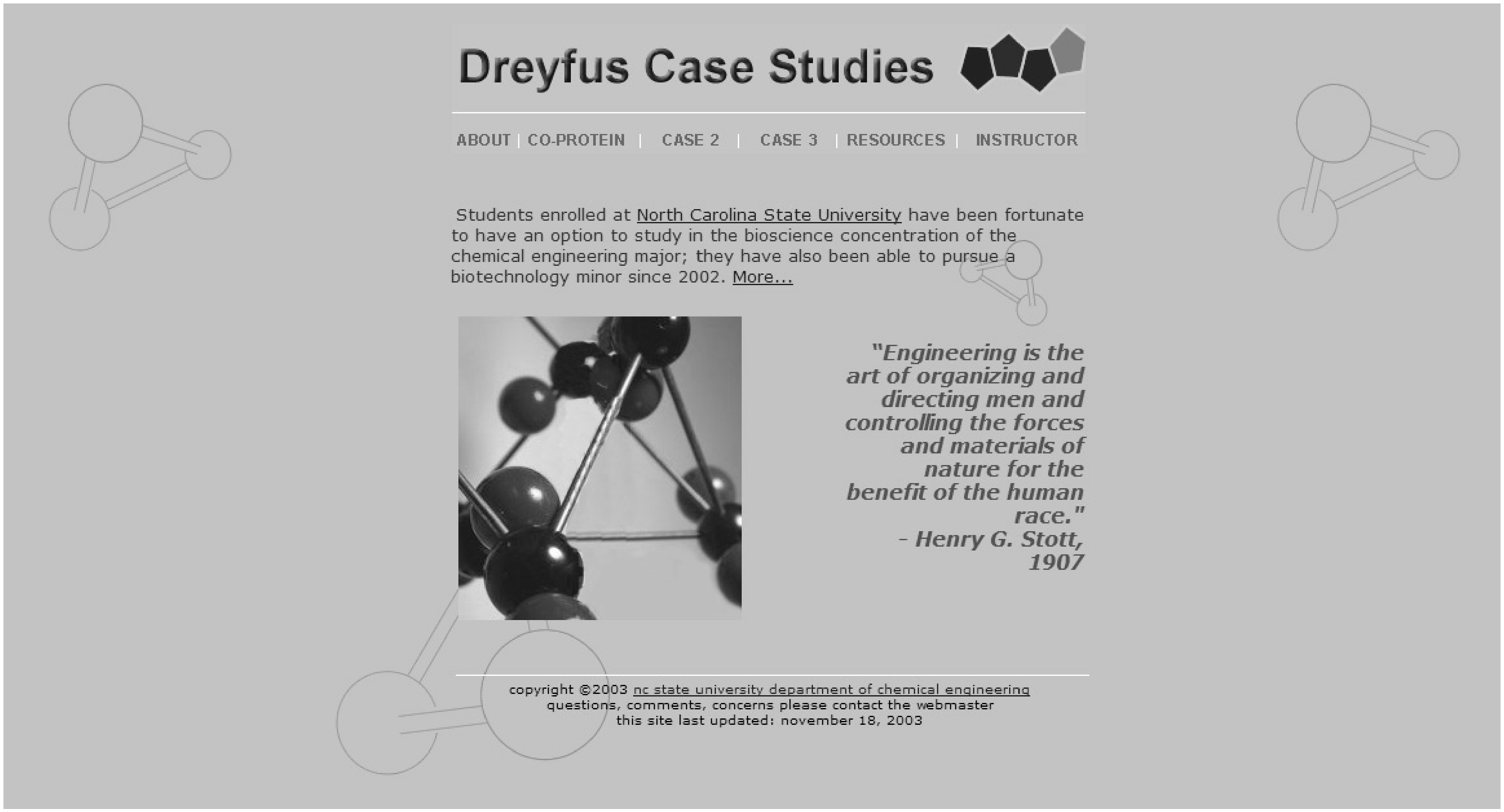
- Purification
- Fermentations
- Facility Design
- Validation

# Website
# Faculty Access




NC STATE UNIVERSITY

# Introductory web page




http://www.ncsu.edu/project/actionagenda/coprotein

NC STATE UNIVERSITY

# Problem Statement

- Retrofit an Existing Biological Facility to Produce Antigenic Co-proteins
  - Potential Candidates:
    - Co-Human Immunodeficiency Virus (HIV)
    - Co-Hepatitis B (Hep C)
    - Co-Hepatitis C (Hep C)
    - Co-Human Papilloma Virus (HPV)
    - Co-Respiratory Syncytial Virus (RSV)
    - Co-Rotavirus

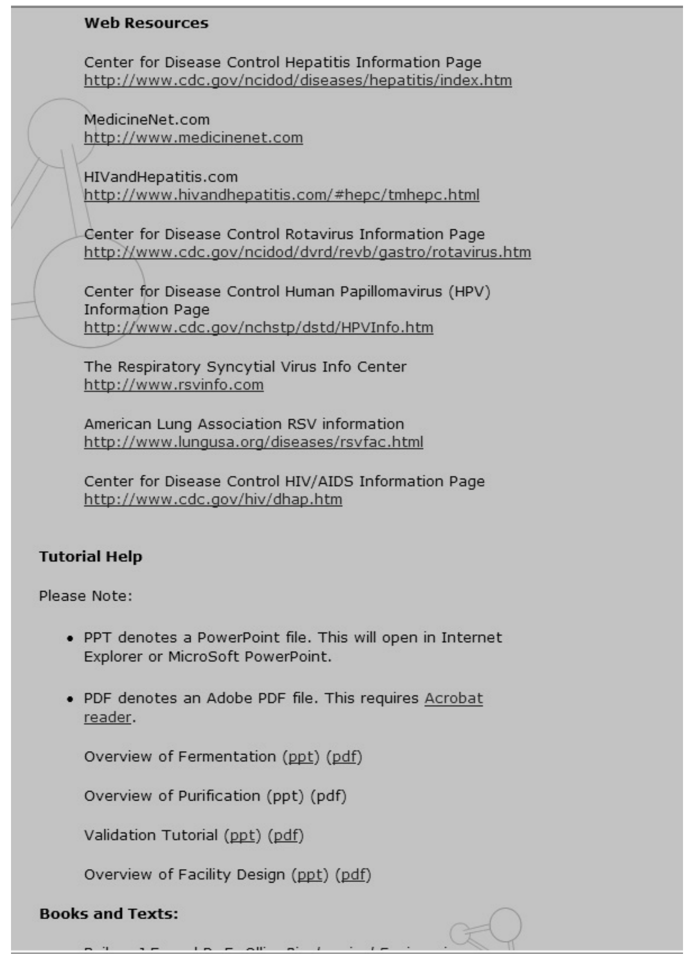
# Problem Statement
# Objectives

- Learning Objectives
  - Manufacturing process
  - Validation practices and techniques
  - Facility design
  - Regulatory obligations
  - Engineering economics
  - Control/Control systems

# Website Resources

Each Case Study has the following resources available to students:

- Web Resources
- Tutorial Help
- Books, Texts, Journals



**Web Resources**

Center for Disease Control Hepatitis Information Page
http://www.cdc.gov/ncidod/diseases/hepatitis/index.htm

MedicineNet.com
http://www.medicinenet.com

HIVandHepatitis.com
http://www.hivandhepatitis.com/#hepc/tmhepc.html

Center for Disease Control Rotavirus Information Page
http://www.cdc.gov/ncidod/dvrd/revb/gastro/rotavirus.htm

Center for Disease Control Human Papillomavirus (HPV) Information Page
http://www.cdc.gov/nchstp/dstd/HPVInfo.htm

The Respiratory Syncytial Virus Info Center
http://www.rsvinfo.com

American Lung Association RSV information
http://www.lungusa.org/diseases/rsvfac.html

Center for Disease Control HIV/AIDS Information Page
http://www.cdc.gov/hiv/dhap.htm

**Tutorial Help**

Please Note:

- PPT denotes a PowerPoint file. This will open in Internet Explorer or MicroSoft PowerPoint.
- PDF denotes an Adobe PDF file. This requires Acrobat reader.

Overview of Fermentation (ppt) (pdf)

Overview of Purification (ppt) (pdf)

Validation Tutorial (ppt) (pdf)

Overview of Facility Design (ppt) (pdf)

**Books and Texts:**

# Information pages

## Example from a tutorial

**Clean in Place (CIP)**: Used to ensure that the production vessels are free of any contaminants. It is essential to ensuring high quality of all finished drug products. Properly automated CIP systems reduce the need for unsafe manual cleaning efforts and increase the level of consistency and effectiveness of the cleaning process.

All are from Pharmaceutical Engineering, the official journal of the ISPE unless noted otherwise. Links can be found at www.ispe.org.


Adams, Dan G. and Deepak Agarwal, 10, (6), 1990, "CIP System Design and Installation", pp 9-15.

Marks, PE, David M., 19, (2), 1999, "An Integrated Approach to CIP/SIP Design for Bioprocess Equipment", pp 34-45.

Sieberling, Dale, 6, (6), 1986, "Clean-In-Place and Sterilize-In-Place Applications in Parenteral Solutions Process", pp 30-35.

# Exemplary Solution Components

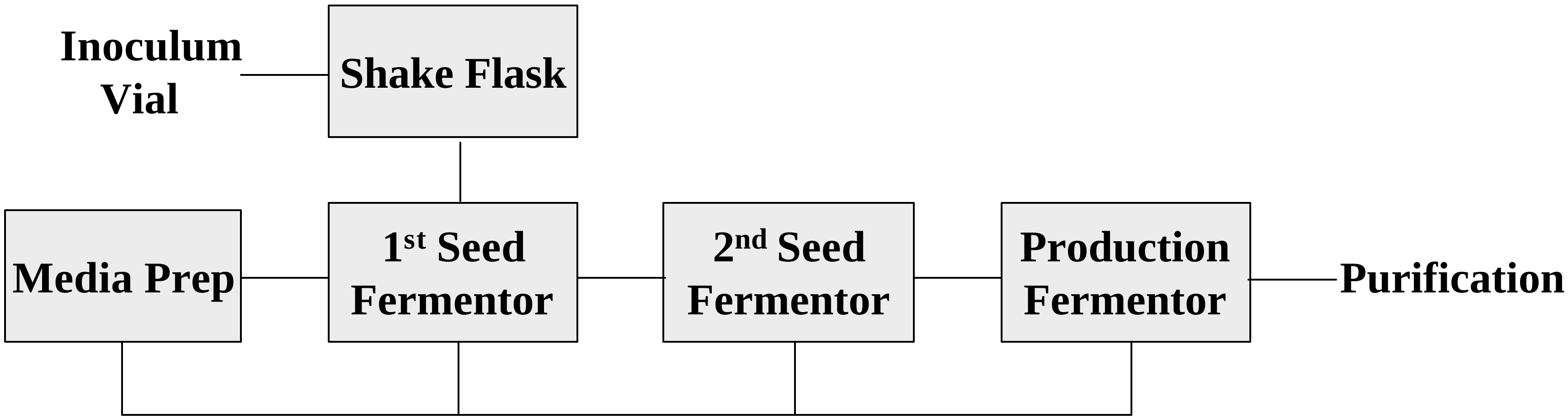
- Executive Summary
- Introduction
- Product Line Determination
- Process Description
- Waste Management
- Regulatory Requirements
- Facility Design
- Validation
- Detailed Manufacturing Costs
- Conclusions

# Exemplary solution
# Recommendations

- Common expression system (*E. coli*)
- Extracellular product
- Analysis restricted to US market
- Market capture grows to 20%
- Subset of potential products selected
  - HIV, HepB, HepC, HPV

# Exemplary Solution Production

- Staged Batch Fermentation
  - Shake Flask
  - Seed Fermentors
  - Production Fermentor



# Exemplary solution
# Purification

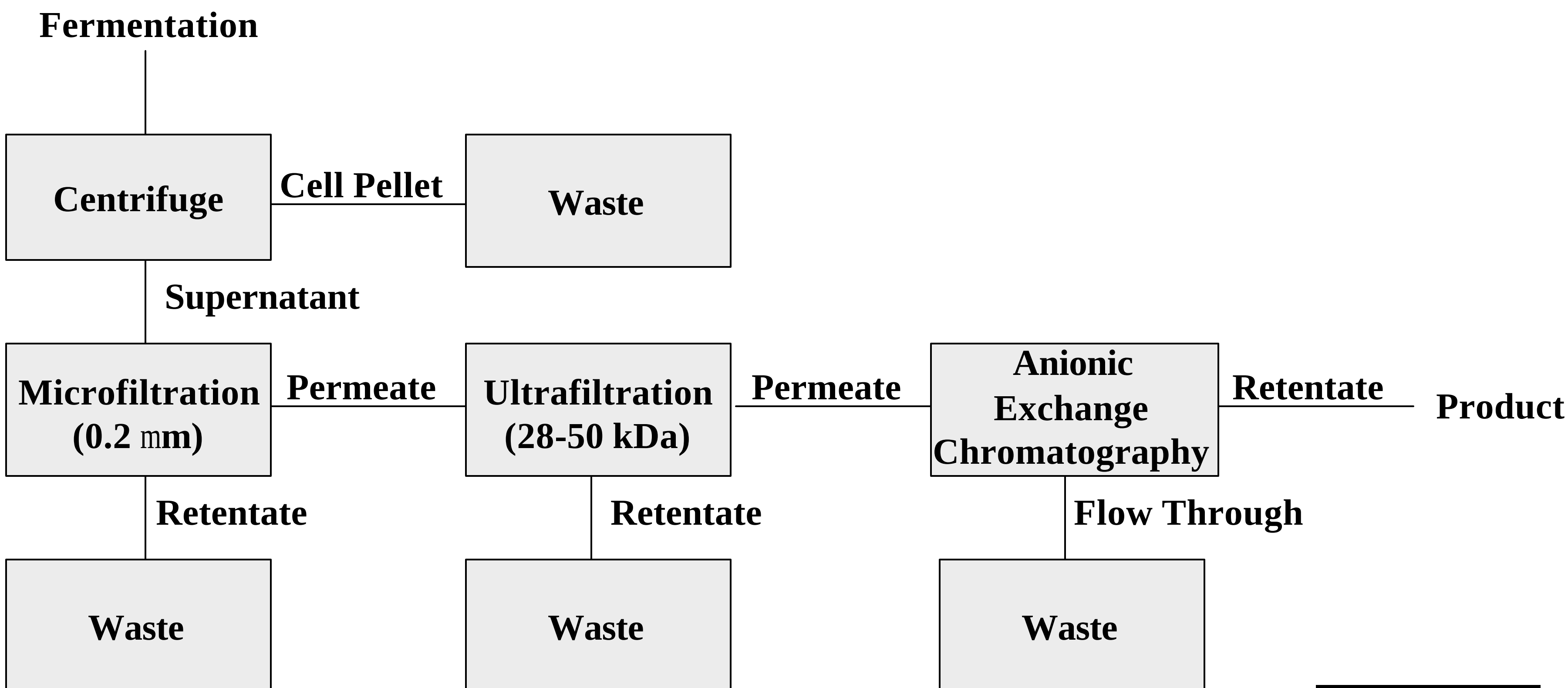
- Similarities
  - Centrifugation
  - 0.2 µm microfiltration
- Differences
  - Ultrafiltration
    - Molecular weight
    - Number of units
  - Chromatography
- Co-HIV is a more complex purification

**Co-Protein Characteristic Table**

| Protein | Molecular Weight (+/- 2 kDa) | Overall Charge | Known Contaminants Co-produced (+/- 2 kDa) |
|---|---|---|---|
| co-HCV | 6 | Neutral | < 3<br>> 10 |
| co-HPV | 10 | Neutral | > 50 |
| co-Rv | 30 | Positive | 28 (-charge)<br>> 50 |
| co-HIV | 10 | Negative | 10 (+ charge)<br>< 50 |

# Exemplary solution
# Purification

## Co-HIV Purification Block Flow Diagram




NC STATE UNIVERSITY

# Economic Analysis
# Fixed Capital Costs

Microsoft Excel - Fixed Capital costs

| | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | Bldg Parameters | | | | | | | | | | | | | | |
| 6 | **Description of Room** | **Length** | **Width** | **Room total** | | | **CSA** | | **HVAC** | | **ELECT** | | **PIPING** | | **I&C** |
| 7 | | **LFT** | | **SF** | | **Unit** | **Costs** | **Unit** | **Costs** | **Unit** | **Costs** | **Unit** | **Costs** | **Unit** | **Costs** |
| 8 | Recovery Room | 39 | 30 | 1170 | | $200 | $234,000 | $75 | $87,750 | $50 | $58,500 | $40 | $46,800 | | W/Equip |
| 9 | Fermentation Room | 25 | 30 | 750 | | $200 | $150,000 | $75 | $56,250 | $50 | $37,500 | $40 | $30,000 | | W/Equip |
| 10 | Media Room | 25 | 25 | 625 | | $10 | $6,250 | $10 | $6,250 | $10 | $6,250 | $10 | $6,250 | | W/Equip |
| 11 | Dirty Equipment | 38 | 25 | 950 | | $5 | $4,750 | $0 | $0 | $0 | $0 | $10 | $9,500 | | W/Equip |
| 12 | Washer | 10 | 17 | 170 | | $0 | $0 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 13 | A/L 1 | 10 | 13 | 130 | | $50 | $6,500 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 14 | Clean Equipment | 30 | 20 | 600 | | $0 | $0 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 15 | Autoclave | 30 | 10 | 300 | | $0 | $0 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 16 | A/L 2 | 10 | 18 | 180 | | $50 | $9,000 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 17 | Gown/Degown | 10 | 24 | 240 | | $50 | $12,000 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 18 | A/L 3 | 10 | 13 | 130 | | $50 | $6,500 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 19 | Gown | 8 | 8 | 64 | | $50 | $3,200 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 20 | Buffer | | | 1187 | | $10 | $11,870 | $0 | $0 | $5 | $5,935 | $10 | $11,870 | $10 | $11,870 |
| 21 | Degown | 11 | 12 | 132 | | $50 | $6,600 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 22 | Storage | 28 | 19 | 532 | | $20 | $10,640 | $5 | $2,660 | $5 | $2,660 | $0 | $0 | | N/A |
| 23 | A/L 4 | 11 | 11 | 121 | | $50 | $6,050 | $0 | $0 | $0 | $0 | $0 | $0 | | N/A |
| 24 | SIP, CIP, Tank storage | | | 1427 | | $50 | $71,350 | $5 | $7,135 | $4 | $5,708 | $25 | $35,675 | | N/A |
| 25 | Fill | 20 | 30 | 600 | | $50 | $30,000 | $20 | $12,000 | $30 | $18,000 | $40 | $24,000 | | W/Equip |
| 26 | Cold Room | 14 | 30 | 420 | | $50 | $21,000 | $20 | $8,400 | $30 | $12,600 | $40 | $16,800 | | W/Equip |
| 27 | Purification 2 | 17 | 30 | 510 | | $150 | $76,500 | $40 | $20,400 | $30 | $15,300 | $40 | $20,400 | | W/Equip |
| 28 | Purification 1 | 51 | 30 | 1530 | | $200 | $306,000 | $75 | $114,750 | $50 | $76,500 | $75 | $114,750 | | W/Equip |
| 29 | | | | | | | **$972,210** | | **$315,595** | | **$238,953** | | **$316,045** | | **$11,870** |
| 30 | | | | **11,768** | | | | | | | | | | | |
| 31 | **Building Costs** | | | | **$1,854,673** | | | | | | | | | | |

NC STATE UNIVERSITY

# Economic Analysis
## Projected Sales Revenue

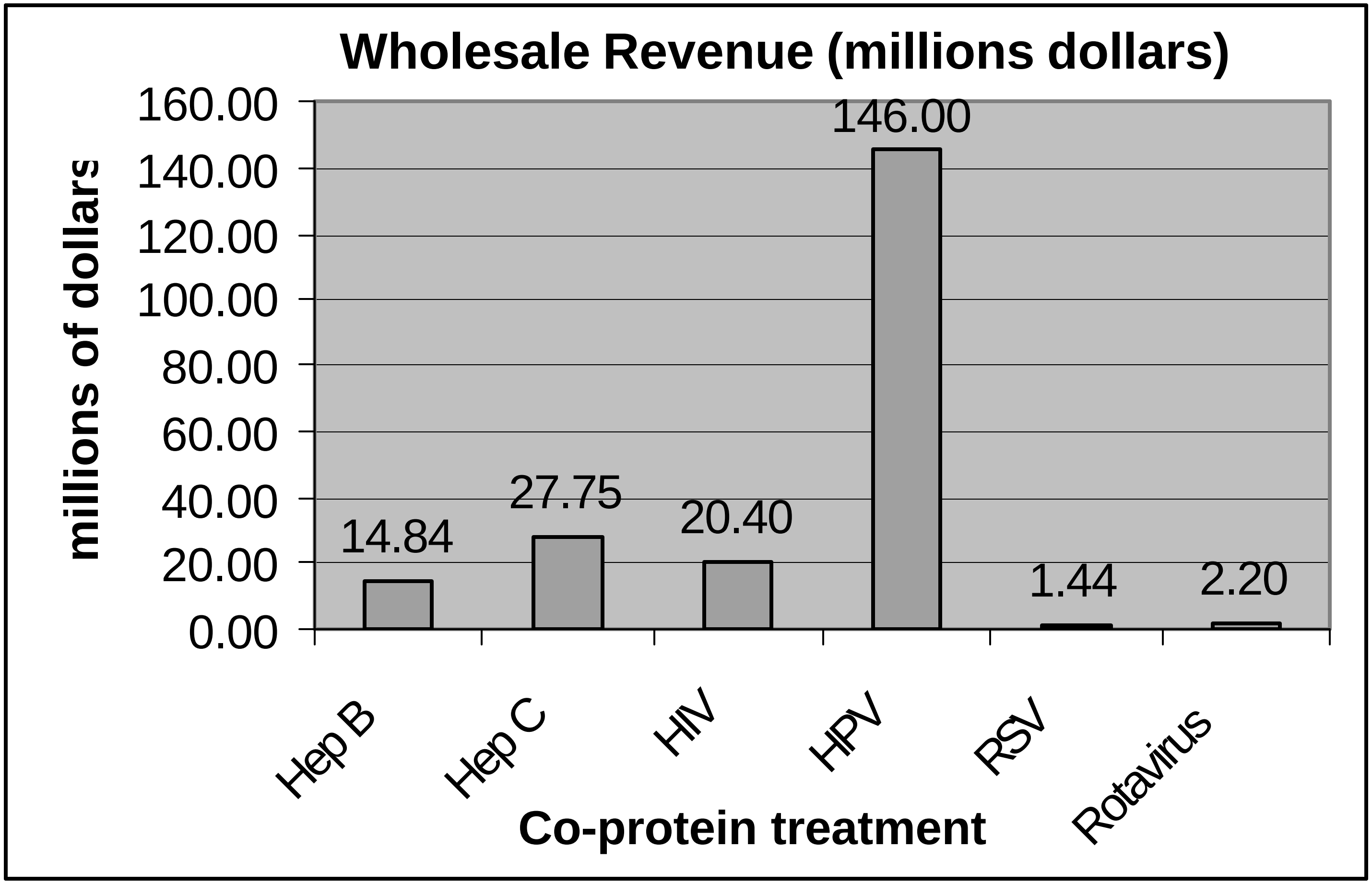



***Figure 7.*** Sales Revenue for year 2006.

# What's next?

- Completion of co-protein case study
  - Implementation advice
  - Expand tutorials
  - Develop SuperPro Designer simulation
  - Testing this year - looking for volunteers

# What's next?

- Ammonia synthesis plant
  - Available Online in February 2004
- Citric acid retrofit/expansion project
  - Expand tutorials
  - Add mechanical engineering component to solution
  - Develop instructor's manual
  - Testing next spring
  - Available Online May 2004

# Acknowledgements

- Dreyfus Foundation
- Alan Bream, Claude Burgess & Larry James (CRB USA)
- Bob Perrone (Fluor Daniel)
- Joel Youngblood (Bayer)
- CHE 451 Students (2001-2003)

# Contact Information

- Website Location:
  http://www.ncsu.edu/project/actionagenda/coprotein
- NC State Chemical Engineering Department
  http://www.che.ncsu.edu


- Dr. Lisa Bullard
  (Lisa_Bullard@ncsu.edu)
- Dr. Steven Peretti
  (peretti@eos.ncsu.edu)
- Patty Niehues
  (pniehues@nc.rr.com)
- Shannon White
  (Shannon_White@ncsu.edu)


NC STATE UNIVERSITY

# ChE's Teaching Introductory Computing to ChE Students -- A Modern Computing Course with Emphasis on Problem Solving and Programming

David E. Clough

Department of Chemical Engineering
University of Colorado
Boulder, CO 80309-0424

Abstract

An easy recipe for fomenting heated debate among ChE faculty is to inject the topic of introductory computing for ChE students into a discussion. Most faculty will have strong opinions that are only muted by the threat of a teaching assignment to such a course. There are many questions: Who should teach introductory computing to our students? Faculty from computer science, a general engineering department, the ChE department, or others? What should be taught? Traditional programming with Fortran, object-oriented programming with C++, problem solving with tools such as Excel and Mathcad, or various mixtures?

In ChE at the University of Colorado, we are no different from many other institutions in that these debates have raged on for decades and continue today. In fact, over the years, our students have taken courses in all the various categories mentioned above. Recently, however, we have settled on a scheme and a course design that is working particularly well and bears consideration by others.

In engineering here, introductory computing is taught under an umbrella course number (GEEN 1300 Introduction to Engineering Computing, 3 credit hours). The various engineering degree programs (chemical, mechanical, civil, architectural, environmental, aerospace) each teach a section of this course, and these sections take on "flavors" according to the preferences of the particular program. Students in electrical engineering, computer engineering, and computer science do not take this course, rather a typical "CS101" course based on C/C++. A significant fraction of the entering students, typically 30%, are "open option," not having declared an engineering major. These students are included in the sections of the GEEN 1300 based on their interest in and leanings toward an engineering major. Also, there has occasionally been an additional section of the course for "open option" students and students not yet in the College of Engineering.

Two years ago, ChE at Colorado initiated a change in this course, taking it away from its traditional Fortran/Excel base. In this transition, two central themes were preserved: scientific/engineering problem solving and structured programming. The new course is divided into four roughly-equal parts. First comes a segment on engineering problem solving using the

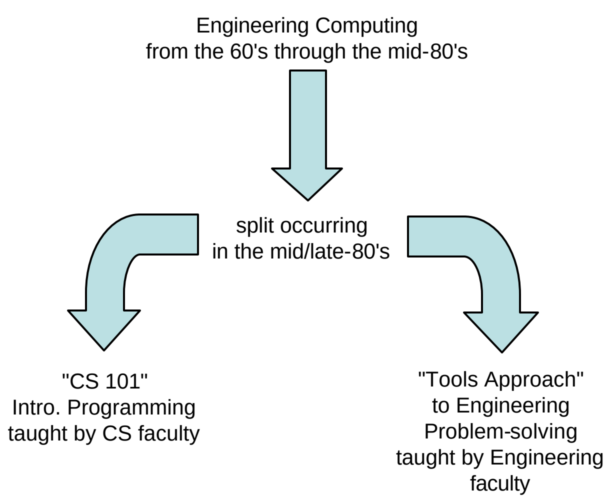
Mathcad package. This is followed by a segment on engineering problem solving and elementary numerical methods with Excel. The third segment expands the use of Excel by introducing structured programming via its Visual Basic for Applications (VBA) language. And the final segment continues the themes with the Matlab package along with an introduction to vector/matrix calculations. Problems from chemistry, physics, engineering and ChE are used throughout.

There are some pedagogical keys to the success of this course. The combined use of Excel and VBA has a strong supporting case. Providing students with knowledge and skills that they can use immediately, during the same semester, in other courses and activities is important to student motivation. Providing a gateway to subsequent use of the software tools and, for some students, to building their computing knowledge in follow-on courses completes the picture. This paper and presentation will provide details of the course design, its evolution, and its evaluation by students and external examiners.

Evolution of engineering computing – from “Slide Rule 100” to “CS 101”

From the 1950’s through the 1960’s in the US, traditional introductory engineering courses that focused on calculations with slide rules and log tables and graphical representation of information were modified to include scientific numerical computer programming. The vehicle for the latter was Fortran programming on mainframe computers using punched cards. The computing component of these courses grew through the 70’s with more attention paid to numerical methods. The minicomputer was a common vehicle, and video terminals gradually replaced the use of punched cards. In the 1980’s technology shifted to personal computers with magnetic storage media. Although there were “experiments” or “movements” with different programming languages during the 70’s and 80’s, such as Pascal, Fortran, in its evolving versions, remained the primary software vehicle.

From the mid-80’s forward through the 90’s, a split occurred in the direction taken toward introductory computing for engineers. In part, this was catalyzed by the growth of computer science programs and the computer science faculty taking over this instructional role. This is illustrated in the figure below.



Both branches are prevalent today. The "CS 101" courses have migrated through several programming languages: Pascal, C, C++, and Java. In the engineering branch, software vehicles such as spreadsheets (first Lotus 123, then Quattro Pro, and now Excel), TK Solver, Mathcad, and Matlab have gradually pushed out programming languages, primarily Fortran, which are becoming endangered species in these courses.

The "CS 101" branch would claim a number of reasons for existence:

- engineers should learn fundamental concepts of programming and computer science
- computing should be taught by computer scientists, not engineers
- engineering faculty are not interested in teaching computing to their students
- these courses provide a significant chunk of student credit hours (SCH) and budgetary resources

as would the "Tools Approach" branch:

- engineering students need a solid grounding in problem solving with modern computing tools
- engineering students need the knowledge and tools required in their professions
- engineering computing and problem-solving are best taught by engineers
- these courses provide a significant chunk of student credit hours (SCH) and budgetary resources

A real assessment reveals that the two branches are complementary and that many/most engineering students can benefit from both courses, although most curricula are found to be too congested to make room for both. That raises the question: *Is each branch at fault more for what it leaves out than what it includes?*

Student needs – a balancing act

Having participated in discussions on introductory computing with engineering faculty over the past 35 years, I am concerned with arguments put forth over and over again. When it comes to computing needs, faculty often confuse what is important for their students with what is important for themselves. Faculty needs, more often than not, align with their research interests and activities, and these are distant, if not disconnected, from the needs of their undergraduate students. Also, faculty often carry an impression of what the needs of professionals are that is off base, by being either out of date or out of touch. Few discussions proceed on the basis of evidence from alumni and employer surveys. Finally, computing is not part of the daily professional existence of most faculty and is not expected to be. Their computing skills are oxidized, and most of their computing is carried out in the trenches by their students. This is, perhaps, part of the more general dilemma of faculty preparing students for a profession that many/most faculty themselves have not practiced.

Even given this troubling backdrop of faculty perspective, an encompassing view of student needs includes several areas that compete for their slice of the instructional pie:

- fundamental knowledge of computing, programming and computers
- awareness of and preparation in emerging aspects of computing
- computing requirements in the other courses of their curriculum
- knowledge and skills required by engineers in their day-to-day professional lives
- opening the door for further study and specialization in computing and computer science

The argument for fundamental knowledge is sound. Such knowledge in computing will transcend the skills and tools of the day. Fundamentals provide the foundation. A criticism is that fundamentals tend to be abstract and difficult for students to grasp and appreciate. Most students learn better inductively, generalizing fundamentals from specific and practical exercises and examples.

Given the rapid rate of change in computing, teaching only the state of the art in the profession will leave students out of date by the time they get there. Therefore, educating them in emerging trends is important. A problem with this is judging which trends will stick and which will be flashes in the pan.

Students will appreciate and be motivated by the acquisition of skills that they can put to immediate use, even during the semester in which they are taking the introductory computing course. Of course, focusing too much on immediate needs may miss the mark when it comes to professional needs. Many computing tools used in the curriculum satisfy learning objectives but are of little use in professional life.

Teaching students in the context of computing vehicles used by practicing professionals has attractive payouts down the road. Focusing on the day-to-day problem solving activities of engineering professionals has high relevance and importance. However, there are difficulties. The learning curve for some software packages is far too steep, and some packages have a knowledge prerequisite, especially in mathematics, that far outstrips the abilities of first-year students.

There is also a significant problem with using tools that automate tasks, where the user has little view of the internal operations of the task – the *black-box* syndrome. This works fine and is greatly appreciated by the professional that already has knowledge of and experience with the task being performed. But there is a great temptation for mindless button-pushing (or mouse-clicking) by students who should be learning what is behind the button. Also, there is the danger of becoming trapped by the built-in capabilities of one's software, in other words, being incapable of extending the software capabilities through programming.

A few students will want to specialize in computing. For example, some students will become software developers in the context of engineering applications. These individuals will require more education in computing, perhaps a minor or even a double degree. The introductory computing course in engineering should not attempt to redirect these students away from computer science; rather, it should open the door.
Since valid arguments can be made for the five areas of need listed above, it becomes a challenge to design an introductory computing experience that balances and, at least in part, satisfies the needs. We have attempted to meet that challenge at the University of Colorado.

Course objectives & outline – narrowing the focus

In the mid-80's, at the University of Colorado, the split in courses between "introductory engineering computing" and "introductory computer science" took place. Since then, two courses have predominated

- GEEN 1300 Introduction to Engineering Computing (3 semester credit hours)
- CSCI 1300 Computer Science 1: Programming (4 semester credit hours)

The engineering computing course is taken by all chemical, civil, environmental and mechanical engineering students. The computer programming course is taken by all electrical engineering and computer science students. Many students take both courses – the courses are considered to be complementary. Computer Science offers a second introductory course

- CSCI 2270 Computer Science 2: Data Structures (4 semester credit hours)

This course is also taken by many engineering students wishing to specialize in computing.

The introductory engineering computing course has been established with a set of objectives that seek to address the needs described above. These are:

1. Problem Solving
   - Apply the "engineering method" to the solution of quantitative problems
   - Evaluate engineering formulas, carrying units and appropriate precision through calculations
   - Practice working in groups to tackle larger-scale engineering problems

2. Symbolic Computing
   - Enter and edit symbolic expressions in computer software
   - Manipulate and solve algebraic expressions
   - Carry out symbolic manipulations for calculus

3. Spreadsheet Techniques
   - Develop efficient spreadsheet skills
   - Set up and interpret "what-if" and case study scenarios
   - Organize and layout spreadsheet solutions to engineering problems

4. Programming Fundamentals
   - Learn how information is represented by different data types
   - Learn program-flow algorithm structure and modularity
   - Program with object-oriented features

5. Elementary Numerical and Statistical Methods
   - Develop the ability to solve single nonlinear algebraic equations using elementary numerical methods, such as bisection, false position or Newton's method
   - Solve sets of linear and nonlinear algebraic equations
   - Carry out regression calculations

6. Software Tools
   - Develop skills with and knowledge of the following software tools:
     Mathcad 2001
     Excel 2000 & Visual Basic for Applications (VBA)
     Matlab 6

The "Problem Solving" objective is a carryover from the old "slide rule" courses. Most entering students lack practice and abilities in numeric problem solving. Many of the lessons from the old courses still have much value and prepare students for the activities in their other courses, in particular, the ChE material & energy balances course. Achieving this objective has an obvious beneficial long-term impact.

"Symbolic Computing" is of immediate utility in the students' math courses. They also make use of this in their science and engineering courses. It is common for students to check their manual work using the computer. A byproduct lesson learned is that not all equations or systems of equations have analytical solutions [not obvious to many freshmen].

"Spreadsheet Techniques" provides a problem-solving methodology that has the broadest and longest impact of any objective in the course. Excel is the day-to-day problem solving tool of most practicing ChE's, and it is the software tool used most frequently by ChE students. Spreadsheet methods are becoming recognized in their own right along with the need to teach them separately to ChE students. The author's AIChE short course in spreadsheet problem-solving has been one of the most frequently offered courses (approaching 100 offerings) over the past dozen years.

"Programming Fundamentals" represents the *lost objective* in many engineering computing courses. It has been retained in this course in a creative way. The fundamentals of structured, algorithmic programming and data structure are introduced via the VBA[1] language within Excel. This provides a natural setting for students to "elevate" prototypes developed on the spreadsheet into more elegant and efficient VBA macros (Subs) and user-defined functions. The portability of the programming concepts is further emphasized by learning the m-script language in Matlab. Achieving this objective opens the door for students in two important ways:

- students who move on to take the computer science courses have a leg up on students with no background in programming – our students enjoy greater success in these follow-on courses

[1] VBA: Visual Basic for Applications, a full-featured, object-oriented programming language with structure superior to C/C++ and Fortran 95.

- programming opens the door to extension of many software packages – without the ability to program, users are forever limited to the conventional, built-in capabilities of the packages

There are certain numerical and statistical methods that represent the bread-and-butter of applications in engineering, both in the academic curriculum and in practice. Several of these are within reach of entering students, and these are represented in the "Elementary Numerical and Statistical Methods" objective. Apart from the practical relevance of equation-solving and regression methods, the lessons learned through a disciplined approach to Gaussian elimination are of great general value.

Our students are exposed to a great variety of software tools during their undergraduate ChE curriculum [Word, Powerpoint, Excel, Mathcad, Matlab, Mathematica, Simulink, Polymath, EZ-Solve, HYSYS, Aspen+, Minitab, Control Station, LabView, LadSim, AutoCAD, to name a few]. To achieve the "Software Tools" objective, we choose to expose the students to several software tools that will be of general utility, teach portable concepts, be accessible and easily acquired, be relevant to their other courses and/or to professional practice. Additionally, we need to prepare the students for the onslaught of the other software packages they will encounter. They obviously need to become skilled at picking up new tools quickly.

The course objectives are embodied in a course outline as follows:

| Topic | No. of Lectures | No. of Labs |
|---|---|---|
| Introduction, problem solving, MathCAD | 6 | 3 |
| Spreadsheet problem solving, Excel | 7 | 4 |
| Introduction to programming, VBA | 7 | 4 |
| Numerical methods, Matlab | 7 | 4 |
| | ---- | ---- |
| | 27 | 15 |

The deliverables of the course are summarized in the grading policy table below.

| | *Points* | *x* | *Weight* | *=* | *Total* | *Pct.* |
|---|---|---|---|---|---|---|
| Project Assignments [9, 1 double] | 100 | | 5 | | 500 | 36.5 |
| Homeworks [4] | 40 | | 2 | | 80 | 5.8 |
| Pop Quizzes [9] | 90 | | 1 | | 90 | 6.6 |
| Lab Sessions [15] | 150 | | 2 | | 300 | 21.9 |
| Midterm Examinations [2] | 200 | | 1 | | 200 | 14.6 |
| Final Examination | 100 | | 2 | | 200 | 14.6 |
| | | | | | ------- | ------- |
| | | | | | 1300 | 100.0 |

Students prefer to have multiple measures of their performance in the course, as opposed to putting all their eggs in a couple baskets. This allows some compromise for students who perform well in an exam setting to those who work better on project assignments.

## Course design – practical considerations

The introductory engineering computing course is taught in several sections that align with the engineering disciplines and the instructors are faculty from those disciplines. Consequently, there is a limited emphasis on examples and problem solving within the particular discipline of the instructor and section. However, the course sections are similar enough that students can cross over sections. This is also important for engineering students who have not declared a specific major yet.

Most believe that a "learn by doing" approach must be an integral part to an introductory computing course. The GEEN 1300 course incorporates a lab component, replacing a 1-hour lecture meeting with a 2-hour workshop in a computer laboratory. The workshops are tutorial in nature with some open-ended, exploratory content. Lab sections are mentored by upper-class undergraduate students who were successful in the course as freshmen. Experience over the years has taught us that this works better than instruction by graduate student TAs, many of whom come to us with varied and limited computing experience.

Outside work is dominated by weekly computing projects that require deliverables of computer files and written reports. There are frequent in-class demonstrations in lieu of conventional lecture. All lecture materials are available to students before class time as Powerpoint files. There are frequent, in-class, "pop" quizzes, two 1-hour midterm examinations, and a 2-1/2-hour final examination.

Nearly all students have their own computers in their dorm rooms, and, although University computer labs are available in many locations for their use around the clock, students prefer to do their work on their own computers. For about 50% of the course, this need is easily answered by Excel, a standard package on their computers. Many students elect to acquire Mathcad for a student price of about $125. Fewer choose to buy the student edition of Matlab, although many do this later in their academic careers when the software package comes into more frequent use.

From our alumni and employer surveys, we find that Mathcad and Matlab are not generally available to practicing ChE's. Of course, Excel is available to all. So, the former packages answer mainly educational and academic needs.

## An example of the pedagogical approach used in the course

The engineering computing course in ChE at Colorado introduces students to various bread-and-butter numerical methods. This is done in a "crawl-then-walk-then-run" fashion where students first solve problems with pencil and paper ("crawl"), then use the spreadsheet environment ("walk"), then program the method in VBA and later in Matlab ("run"), and finally use the "black box" capabilities of the software packages (Mathcad, Excel & Matlab). This approach has two benefits:

- students gain an appreciation and understanding of the method and its limitations that is not possible by merely pushing the buttons of the "black box" software features, and
- students learn the discipline required to understand and carry out an algorithmic numerical method.

Methods taught include equation solving (single nonlinear equations, methods such as bisection, false position and Newton's), solving sets of linear algebraic equations via Gaussian elimination, and linear regression. We will illustrate the approach here with the Gaussian elimination method.

For a simple set of three linear algebraic equations in three unknowns, students are taught the naïve Gaussian elimination method. They then must practice the method by hand and complete an in-class "pop" quiz to test their knowledge. See the figure below.

Set of equations

$$x_1 + 2x_2 + 3x_3 = 0$$
$$4x_1 + 5x_2 + 6x_3 = 1$$
$$7x_1 + 8x_2 + 8x_3 = -1$$

Coefficients & constants

$$\begin{array}{cccc} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 1 \\ 7 & 8 & 8 & -1 \end{array}$$

1) Normalize 1,1
already done!

2) Reduce 2,1
row 2 − 4 · row 1 → row 2

$$\begin{array}{rcccc} & 4 & 5 & 6 & 1 \\ (-) & 4 & 8 & 12 & 0 \\ \hline & 0 & -3 & -6 & 1 \end{array}$$

3) Reduce 3,1
row 3 − 7 · row 1 → row 3

$$\begin{array}{rcccc} & 7 & 8 & 8 & -1 \\ (-) & 7 & 14 & 21 & 0 \\ \hline & 0 & -6 & -13 & -1 \end{array}$$

Current matrix

$$\begin{array}{cccc} 1 & 2 & 3 & 0 \\ 0 & -3 & -6 & 1 \\ 0 & -6 & -13 & -1 \end{array}$$

4) Normalize 2,2

$$\begin{array}{cccc} 0 & 1 & 2 & -1/3 \end{array}$$

5) Reduce 3,2
row 3 − (−6) · row 2 → row 3

$$\begin{array}{rcccc} & 0 & -6 & -13 & -1 \\ (-) & 0 & -6 & -12 & 2 \\ \hline & 0 & 0 & -1 & -3 \end{array}$$

6) Normalize 3,3

$$\begin{array}{cccc} 0 & 0 & 1 & 3 \end{array}$$

$(x_3 = 3)$

Current matrix

$$\begin{array}{cccc} 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & -1/3 \\ 0 & 0 & 1 & 3 \end{array}$$

end of forward pass

7) Reduce 2,3
row 2 − 2 · row 3 → row 2

$$\begin{array}{rcccc} & 0 & 1 & 2 & -1/3 \\ (-) & 0 & 0 & 2 & 6 \\ \hline & 0 & 1 & 0 & -19/3 \end{array}$$

$(x_2 = -19/3)$

8) Reduce 1,3
row 1 − 3 · row 3 → row 1

$$\begin{array}{rcccc} & 1 & 2 & 3 & 0 \\ (-) & 0 & 0 & 3 & 9 \\ \hline & 1 & 2 & 0 & -9 \end{array}$$

Current matrix

$$\begin{array}{cccc} 1 & 2 & 0 & -9 \\ 0 & 1 & 0 & -19/3 \\ 0 & 0 & 1 & 3 \end{array}$$

9) Reduce 1,2
row 1 − 2 · row 2 → row 1

$$\begin{array}{rcccc} & 1 & 2 & 0 & -9 \\ (-) & 0 & 2 & 0 & -38/3 \\ \hline & 1 & 0 & 0 & 11/3 \end{array}$$

$(x_1 = 11/3)$

Final matrix

$$\begin{array}{cccc} 1 & 0 & 0 & 11/3 \\ 0 & 1 & 0 & -19/3 \\ 0 & 0 & 1 & 3 \end{array}$$

Solution

$x_1 = 11/3$, $x_2 = -19/3$, $x_3 = 3$

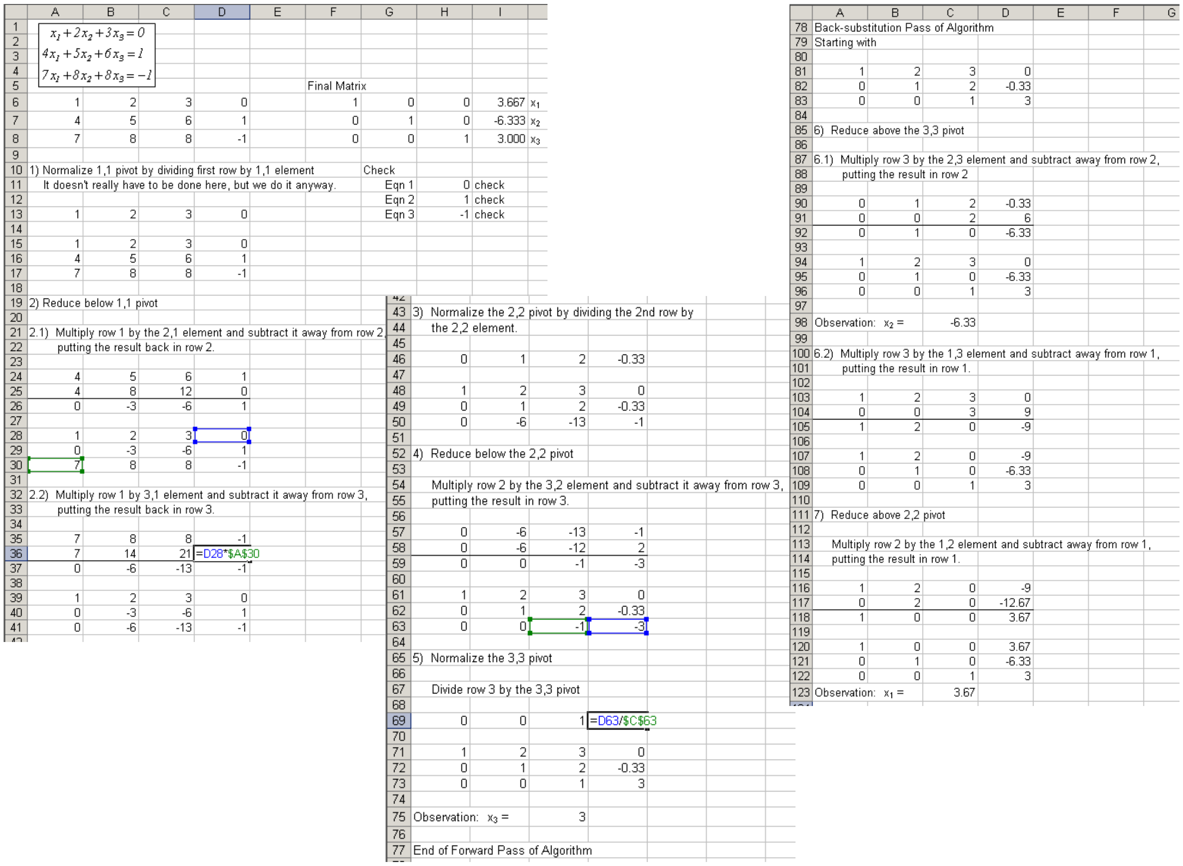
Check

Eqn 1: $11/3 + 2(-19/3) + 3(3) \stackrel{?}{=} 0$
$11/3 - 38/3 + 27/3 \stackrel{?}{=} 0$
$0 = 0$ ✓

Eqn 2: $4(11/3) + 5(-19/3) + 6(3) \stackrel{?}{=} 1$
$44/3 - 95/3 + 54/3 \stackrel{?}{=} 1$
$1 = 1$ ✓

Eqn 3: $7(11/3) + 8(-19/3) + 8(3) \stackrel{?}{=} -1$
$77/3 - 152/3 + 72/3 \stackrel{?}{=} -1$
$-1 = -1$ ✓

Following the manual practice, students implement the procedure on an Excel spreadsheet with formulas as shown below.



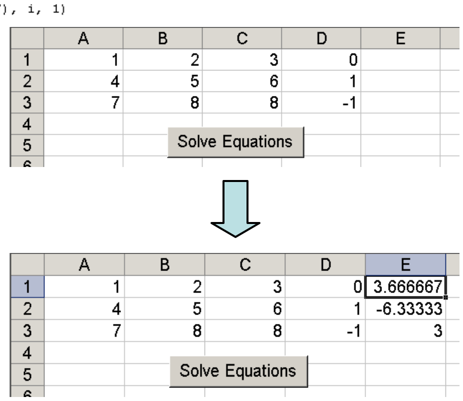
| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $x_1+2x_2+3x_3=0$ | | | | | | | | |
| 3 | $4x_1+5x_2+6x_3=1$ | | | | | | | | |
| 4 | $7x_1+8x_2+8x_3=-1$ | | | | | | | | |
| 5 | | | | | | Final Matrix | | | |
| 6 | 1 | 2 | 3 | 0 | | 1 | 0 | 0 | 3.667 $x_1$ |
| 7 | 4 | 5 | 6 | 1 | | 0 | 1 | 0 | -6.333 $x_2$ |
| 8 | 7 | 8 | 8 | -1 | | 0 | 0 | 1 | 3.000 $x_3$ |
| 10 | 1) Normalize 1,1 pivot by dividing first row by 1,1 element | | | | | | Check | | |
| 11 | It doesn't really have to be done here, but we do it anyway. | | | | | | Eqn 1 | 0 | check |
| 12 | | | | | | | Eqn 2 | 1 | check |
| 13 | 1 | 2 | 3 | 0 | | | Eqn 3 | -1 | check |
| 15 | 1 | 2 | 3 | 0 | | | | | |
| 16 | 4 | 5 | 6 | 1 | | | | | |
| 17 | 7 | 8 | 8 | -1 | | | | | |
| 19 | 2) Reduce below 1,1 pivot | | | | | | | | |
| 21 | 2.1) Multiply row 1 by the 2,1 element and subtract it away from row 2, | | | | | | | | |
| 22 | putting the result back in row 2. | | | | | | | | |
| 24 | 4 | 5 | 6 | 1 | | | | | |
| 25 | 4 | 8 | 12 | 0 | | | | | |
| 26 | 0 | -3 | -6 | 1 | | | | | |
| 28 | 1 | 2 | 3 | 0 | | | | | |
| 29 | 0 | -3 | -6 | 1 | | | | | |
| 30 | 7 | 8 | 8 | -1 | | | | | |
| 32 | 2.2) Multiply row 1 by 3,1 element and subtract it away from row 3, | | | | | | | | |
| 33 | putting the result back in row 3. | | | | | | | | |
| 35 | 7 | 8 | 8 | -1 | | | | | |
| 36 | 7 | 14 | 21 | =D28*$A$30 | | | | | |
| 37 | 0 | -6 | -13 | -1 | | | | | |
| 39 | 1 | 2 | 3 | 0 | | | | | |
| 40 | 0 | -3 | -6 | 1 | | | | | |
| 41 | 0 | -6 | -13 | -1 | | | | | |

| | A | B | C | D |
|---|---|---|---|---|
| 43 | 3) Normalize the 2,2 pivot by dividing the 2nd row by | | | |
| 44 | the 2,2 element. | | | |
| 46 | 0 | 1 | 2 | -0.33 |
| 48 | 1 | 2 | 3 | 0 |
| 49 | 0 | 1 | 2 | -0.33 |
| 50 | 0 | -6 | -13 | -1 |
| 52 | 4) Reduce below the 2,2 pivot | | | |
| 54 | Multiply row 2 by the 3,2 element and subtract it away from row 3, | | | |
| 55 | putting the result in row 3. | | | |
| 57 | 0 | -6 | -13 | -1 |
| 58 | 0 | -6 | -12 | 2 |
| 59 | 0 | 0 | -1 | -3 |
| 61 | 1 | 2 | 3 | 0 |
| 62 | 0 | 1 | 2 | -0.33 |
| 63 | 0 | 0 | -1 | -3 |
| 65 | 5) Normalize the 3,3 pivot | | | |
| 67 | Divide row 3 by the 3,3 pivot | | | |
| 69 | 0 | 0 | 1 | =D63/$C$63 |
| 71 | 1 | 2 | 3 | 0 |
| 72 | 0 | 1 | 2 | -0.33 |
| 73 | 0 | 0 | 1 | 3 |
| 75 | Observation: $x_3$ = | | 3 | |
| 77 | End of Forward Pass of Algorithm | | | |

| | A | B | C | D |
|---|---|---|---|---|
| 78 | Back-substitution Pass of Algorithm | | | |
| 79 | Starting with | | | |
| 81 | 1 | 2 | 3 | 0 |
| 82 | 0 | 1 | 2 | -0.33 |
| 83 | 0 | 0 | 1 | 3 |
| 85 | 6) Reduce above the 3,3 pivot | | | |
| 87 | 6.1) Multiply row 3 by the 2,3 element and subtract away from row 2, | | | |
| 88 | putting the result in row 2 | | | |
| 90 | 0 | 1 | 2 | -0.33 |
| 91 | 0 | 0 | 2 | 6 |
| 92 | 0 | 1 | 0 | -6.33 |
| 94 | 1 | 2 | 3 | 0 |
| 95 | 0 | 1 | 0 | -6.33 |
| 96 | 0 | 0 | 1 | 3 |
| 98 | Observation: $x_2$ = | | -6.33 | |
| 100 | 6.2) Multiply row 3 by the 1,3 element and subtract away from row 1, | | | |
| 101 | putting the result in row 1. | | | |
| 103 | 1 | 2 | 3 | 0 |
| 104 | 0 | 0 | 3 | 9 |
| 105 | 1 | 2 | 0 | -9 |
| 107 | 1 | 2 | 0 | -9 |
| 108 | 0 | 1 | 0 | -6.33 |
| 109 | 0 | 0 | 1 | 3 |
| 111 | 7) Reduce above 2,2 pivot | | | |
| 113 | Multiply row 2 by the 1,2 element and subtract away from row 1, | | | |
| 114 | putting the result in row 1. | | | |
| 116 | 1 | 2 | 0 | -9 |
| 117 | 0 | 2 | 0 | -12.67 |
| 118 | 1 | 0 | 0 | 3.67 |
| 120 | 1 | 0 | 0 | 3.67 |
| 121 | 0 | 1 | 0 | -6.33 |
| 122 | 0 | 0 | 1 | 3 |
| 123 | Observation: $x_1$ = | | 3.67 | |

Later, students "elevate" their spreadsheet solution into VBA within Excel in the form of a macro (Sub). This includes flow-charting the Gaussian elimination method first. The VBA code is shown in the figure below.

```
Option Explicit
Option Base 1
Sub Gauss()
  Dim Amat() As Double, bvec() As Double
  Dim i As Integer, j As Integer, k As Integer
  Dim n As Integer
  n = Range("A").Rows.Count
  ReDim Amat(n, n) As Double
  ReDim bvec(n) As Double
  For i = 1 To n
    For j = 1 To n
      Amat(i, j) = Application.WorksheetFunction.Index(Range("A"), i, j)
    Next j
  bvec(i) = Application.WorksheetFunction.Index(Range("b"), i, 1)
  Next i
  For i = 1 To n
    For j = i + 1 To n
      Amat(i, j) = Amat(i, j) / Amat(i, i)
    Next j
    bvec(i) = bvec(i) / Amat(i, i)
    For k = i + 1 To n
      For j = i + 1 To n
        Amat(k, j) = Amat(k, j) - Amat(i, j) * Amat(k, i)
      Next j
      bvec(k) = bvec(k) - bvec(i) * Amat(k, i)
    Next k
  Next i
  For i = n To 2 Step -1
    For j = i - 1 To 1 Step -1
      bvec(j) = bvec(j) - Amat(j, i) * bvec(i)
    Next j
  Next i
  Range("b").Select
  ActiveCell.Offset(0, 1).Select
  For i = 1 To n
    ActiveCell.Offset(i - 1, 0).Value = bvec(i)
  Next i
End Sub
```



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 0 | |
| 2 | 4 | 5 | 6 | 1 | |
| 3 | 7 | 8 | 8 | -1 | |
| 4 | | | | | |
| 5 | | Solve Equations | | | |

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 0 | 3.666667 |
| 2 | 4 | 5 | 6 | 1 | -6.33333 |
| 3 | 7 | 8 | 8 | -1 | 3 |
| 4 | | | | | |
| 5 | | Solve Equations | | | |

Subsequently, by testing another set of equations, students "discover" the pivoting limitation with the naïve Gaussian method and they implement a partial pivoting feature. This introduces them to iterative refinement of numerical methods. During the Matlab segment of the course, they translate their VBA code to a Matlab m-script, which is shown below.

```
function soln = GaussElim(Amat,bvec)
n = length(bvec);
eps = 1.e-6;
sing = 0;
for i = 1:n
   [Amat bvec] = Pivot(Amat,bvec,n,i);
   if abs(Amat(i,i))<eps
      sing = 1;
      break
   end
   Amat(i,i+1:n)=Amat(i,i+1:n)/Amat(i,i);
   bvec(i) = bvec(i)/Amat(i,i);
   for k = i+1:n
      Amat(k,i+1:n) = Amat(k,i+1:n) - Amat(i,i+1:n) .* Amat(k,i);
      bvec(k) = bvec(k)-bvec(i)*Amat(k,i);
   end
end
if sing == 0
   for i = n:-1:2
      bvec(1:i-1) = bvec(1:i-1) - bvec(i) * Amat(1:i-1,i);
   end
   soln = bvec;
else
   disp('system of equations is singular')
end
```

```
function [x,y] = Swap(a,b)
x=b;
y=a;
```

Finally, students use Mathcad, Excel, and Matlab to solve linear equations in a more streamlined, "black box" fashion. Examples of these solutions are shown below.

**Mathcad solution of linear algebraic equations**

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{pmatrix} \qquad b := \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

$$x := \mathrm{lsolve}(A, b) \qquad x = \begin{pmatrix} 3.667 \\ -6.333 \\ 3 \end{pmatrix}$$

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | | 0 |
| 2 | 4 | 5 | 6 | | 1 |
| 3 | 7 | 8 | 8 | | -1 |
| 4 | | | A | | b |
| 5 | 3.667 | | | | |
| 6 | -6.333 | <== mmult(minverse(A),b) | | | |
| 7 | 3 | | | | |
| 8 | | | | | |
| 9 | Excel solution of linear equations using array functions | | | | |
| 10 | | | | | |

Matlab solution of linear algebraic equations using matrix "left division"

```
>> A=[1 2 3; 4 5 6; 7 8 8]

A =

     1     2     3
     4     5     6
     7     8     8

>> b=[0 1 -1]'

b =

     0
     1
    -1

>> x=A\b

x =

    3.6667
   -6.3333
    3.0000
```

In a subsequent lab project, students apply these techniques to the solution of a problem in a ChE setting.

Progress report – student and external assessment

The Introduction to Engineering Computing course (GEEN 1300) was revised to its current design for the Fall 2000 semester. Student evaluations via the standard University survey have provided the following information:

| Semester | No. of Responses | Course Rating | Instructor Rating | Workload Rating |
|---|---|---|---|---|
| Fall 2000 | 63 | B | B+ | 6.2 |
| Fall 2001 | 52 | B | B+ | 6.8 |

Compared to the history of ratings for this and similar courses, the course and instructor ratings are above average. The average course rating is C and instructor rating is C+. Workload is rated on a 0-to-10 scale with 5 being "about right" in student minds. These ratings are the norm, more than students would like, but not cruel and inhumane.

Observations from student comments and additional surveys are summarized below:

- Students preferred Excel/VBA over Mathcad and Matlab. Mathcad finished a distant 2nd with Matlab close behind.
- In Mathcad, students liked the presentation of equations, symbolic operations, and explicit handling of units. They did not like the finicky editing and the graphics.
- In Excel/VBA, students liked the logical layout of the spreadsheet with everything organized and the results visible. Reviews on programming with VBA were mixed but tilted to the positive. Some thought that VBA could get more complicated than they would like. Students wished that Excel had symbolic capabilities.
- As for Matlab, students liked the vector/matrix capabilities, although they realized that much of this power was beyond their appreciation. Students disliked the Matlab syntax that includes "dots" for array operations. They strongly disliked the Matlab command window interface, considering it primitive when compared to Mathcad and Excel. They really liked Matlab's 3D plotting capabilities. They disliked Matlab's C-like loop structures, seeing them as inferior to those of VBA.
- Over 80% of the students used Excel/VBA in one or more other courses during the same semester of the computing course. About 40% of the students used Mathcad in another course. Only about 10% of the students used Matlab outside of the computing course. Such use was generally spontaneous and not required in these other courses. Students appreciated the immediate impact of their learning.
- In comparison to their other freshman-level courses (calculus, chemistry, physics, etc.), students felt strongly that they learned significantly more in the computing course.

As part of the ABET 2000 process, the course was evaluated by the external Advisory Committee of the Department of Chemical Engineering. Their findings are summarized in the Department's annual ABET report for academic year 2000-2001:

GEEN-1300 – Introduction to Computing
(ABET Score 3.63/4.00; Learning Goals 4.00/4.00))
Reviewers: Ann Butchello, Dynegy Midstream Services, Inc.
Vern Norviel, Affymetrix
Dan Schwartz & Dhinakar Kompala, CU Faculty
Hiwot Molla & Matt Zimmerman, CU Students

1. Good introduction to Excel, which is used extensively after that.
2. Homework requires explanation of approach – not just an answer (Great!)
3. Excellent introduction to concept of engineering and how to think like an engineer.
4. Not clear how much communication or teamwork is used in this course (consider removing this ABET Program Element from the ABET matrix)

The 4th comment above is being addressed in adjustments to the course.

A longer-term evaluation of the course awaits the passage of time.

Should ChE's teach computing to ChE students?

In short, we would claim a resounding "YES!" The benefits are both direct and indirect.

Direct benefits:

- students learn computing skills that are on target for subsequent ChE courses and professional practice,
- students are motivated to learn by exercises and projects with a ChE (and, more generally, engineering) flavor, and
- we can challenge our students more than might be done in other courses that are watered down.

Indirect benefits:

- ChE faculty establish contact with ChE students early on in the curriculum (higher retention in the short term – stronger relations with alumni in the long term),
- ChE faculty are in touch with the academic needs of their entering students, and
- ChE faculty teaching (and assessing) the course provides a tighter and more effective feedback loop vis-à-vis the ABET 2000 continuous improvement process.

ChE faculty from other institutions may express skepticism, claiming that they have no control over the introductory computing course provided to their students. We would suggest that they take control over it. Ultimately, the curriculum offered to ChE students should be controlled by ChE faculty. *Where there is a will, there is a way!* Others may question whether they have faculty who are qualified to teach introductory computing. Of course they have them. It is a matter of one or more faculty with the abilities walking up to the challenge. Bringing new students into the world of engineering and ChE via a course like this is a rewarding experience for a faculty member.

## Bibliography


1. Clough, David E., Steven C. Chapra, and Gary S. Huvard, "A Change in Approach to Engineering Computing for Freshmen – Similar Directions at Three Dissimilar Institutions", 2001 ASEE Annual Conference, Albuquerque, NM.

2. Clough, David E., "In with the new, but not quite out with the old – Introductory Computing for Engineers at the University of Colorado", 2001 ASEE Rocky Mountain Section Annual Meeting, Salt Lake City, UT.

3. Weimer, Alan W., ABET Annual Report – Academic Year 2000-2001, Department of Chemical Engineering, University of Colorado, Boulder, CO.

4. Student Course Evaluation results, GEEN 1300 Introduction to Engineering Computing, University of Colorado, Boulder, CO, http://www.colorado.edu/pba/fcq/by_coll/en/geen.html.


## Author Information


DAVID E. CLOUGH


Dave Clough is Professor of Chemical Engineering at the University of Colorado and has been on the faculty there since 1975. His research interests are centered on the optimization and control of chemical processes. He teaches process control, applied statistics, and introductory computing. He lives in a log home in the Rocky Mountains.
Email: David.Clough@Colorado.edu

# DEVELOPMENT OF AN OPEN SOURCE CHEMICAL PROCESS SIMULATOR


**Raul Cota[1], Marco Satyro[2], Craig Morris[3], Bill Svrcek[1], Brent Young[1]**

**[1] University of Calgary**
**2500 University Dr, NW. T2N 1N4, Calgary, AB, Canada**

**[2] Virtual Materials Group**
**657 Hawkside Mews, NW. T3G 3S1, Calgary, AB, Canada**

**[3] RedTree Development**
**4386 Timberline Cr. V0B 1M6, Fernie, BC, Canada**



## Abstract

Process simulators are a key software tool used in the fluid processing industries for day to day process calculations, for process design, for process optimization and for the debottlenecking of existing processes. Current commercial process simulators do not provide the simulator's source code; the users must rely on a closed black-box approach for the unit process models. This historical approach to software development and usage is diametrically opposite to that of open source software. The open source paradigm provides a near optimal solution for the development of robust and reliable software applications while using minimal formal resources. It is recognized that the open source approach is a feasible software development solution that overcomes some of the shortcomings of the currently available commercial process simulators. This paper presents the development of a state of the art open source process simulator; as well, it also provides sample plant simulations.

## Key words

Process Simulator, Chemical, Open Source, Sim42, Hydrocarbon.

## 1. Introduction

The importance of chemical process simulators is well documented, as they are important tools for modeling plants, while providing opportunities for optimization and debottlenecking of existing processes (Grinthal, 1993). All commercial available process simulators follow the traditional development method, which hides the source code from the users thus relying on a closed, black-box approach.

During the last decade, open source software has grown as a near optimal solution for the development of robust and reliable software applications using minimal formal resources (Moody, 2001). This trend was popularized during the development of the Linux operating system and has since moved towards end user applications. Open source is an innovative solution for the development of a state of the art process simulator that would pool talent from a large audience while providing individual users with an in-depth knowledge of its operation and the freedom to perform any desired modifications without the need to purse special agreements with software providers. This in turn fosters the development of new applications for process simulation originally unseen by the original software developers.

The underling principle of Open Source Software is program code distribution, usually by using Internet based tools, thus permitting it to be shared by all. This allows engineers to use it, test it, and take the development further and faster by providing bug fixes, changes or proposals for enhancements to the community at-large. The success of open source compared to traditional commercial software is that its testing and development base is not constrained to employees within a company, but rather it is in a sense a community driven effort. The open and free nature of such an environment nurtures the involvement of developers and users with many different needs and services varied niche interests that would otherwise be ignored by large commercial software providers.

The design of the simulator follows good software practice in which modules are created with well defined interfaces allowing for parallel and decentralized development. Individual modules can be replaced by special custom modules or by commercially available ones. For example, the simulator has a well-defined thermodynamic interface that allows different property packages to be “plugged” in, such as the commercially available thermodynamic and physical properties engine provided by Virtual Materials Group (1).

## 2. "Traditional" Software practices

In this paper, the term "traditional software practices" refers to software providers that operate in a business model in which the source code of the applications is not released to the users. This "traditional" approach results in a number of shortcomings and limitations in the final products [2].

- Development base is constrained to employees within a company
- Enhancements to the application capture only the company's interpretation of the needs of their current market
- In general, users have to wait a considerable amount of time for new releases
- Specific customizations to the product can only be made by the company that developed the software product, thus creating a dependency. This is a very important factor when dealing with highly specialized and expensive applications.

In the area of chemical process applications, the currently available commercial process simulators, tend to provide a "complete solution" that involves a great deal of specialized technologies and scientific knowledge. These technologies range from highly specific thermodynamic property packages to user friendly graphical interfaces (GUI). This complete solutions approach results in software companies that must invest resources in many areas depending only on their employees and strategic partners to incorporate new developments into their products.

Software companies that focus on a small niche of the process simulation market (thermodynamics for example) may find it difficult to grow the business because they must depend on other available commercial simulators to link to their products and this may pose either technical or economic restrictions. This issue has been addressed in recent years, resulting in open software architecture initiatives (Braunschweig and Pantelides, 1996), where commercial software companies agree to standard application interfaces. “Open Architecture” is a promising generic approach to linking applications (Edwards and Merkel, 1996) without having access to source code.

## 3. Open Source Software (OSS)

Development of OSS is a near optimal solution for the development of robust and reliable software applications using minimal formal resources (Hecker, 1999). The key aspect of OSS is that it releases the code to the users usually via the Internet, at no cost. This allows for the rapid deployment of, testing of, development of by accepting fixes and enhancement proposals or changes.

The term "Open Source" is not controlled or owned by any individual or corporation (Scacchi, 2002). However, the term usually refers to software that is distributed under terms that comply with the Open Source Definition (OSD) [3]. The OSD is maintained by the Open Source Initiative (OSI) [4] and it specifies a few points beyond just allowing access to the code. The OSD covers points related to redistribution of work, derived work, integrity of author's code, discrimination, etcetera. See website [3] for the full text of the OSD.

The OSI maintains a list of OSD compliant licenses and many open source projects just distribute their applications under the terms of one of those previously certified licenses (Wu and Lin, 2001). At the time of this publication the OSI web site lists nearly forty different licenses. Some popular licenses are, the General Public License (GPL), the Lesser GPL (LGPL), the Berkeley Software Distribution (BSD) and the MIT license. For a more complete list of licenses refer to website [5].

## 4. Advantages of OSS

The open and free nature of OSS nurtures the involvement of developers and users with many different needs and the servicing of varied niches interests that would otherwise be ignored by the large commercial software providers. OSS also benefits from rapid releases and prompt feedback because it is not constrained by time zones or work hours. Also, having access to the code results in quick bug fixes that immediately become available to the community at large [2]. An additional important aspect of OSS is that the application does not necessarily depend on the involvement of specific persons or corporations. Everyone has access to the code, hence if the current management resigns, another person or group can continue the project. It has a life of its own!

In chemical engineering, it should be noted that some developments might involve crucial technology that a company can not release to the public domain for strategic or economic reasons. Therefore, an open source process simulator must be designed and licensed in such a way that it contemplates the addition of proprietary as well as open source software applications.

## 5. Organization and the Sim42 foundation

The name of the newly created simulator is Sim42 (REF). The code is resident on a public server and can be accessed via CVS [7], which is also OSS and is the most common tool used for managing OSS projects (Asklund and Bendix, 2002).

Sim42 is licensed using the BSD [8] open source license. This permits the free distribution and modification of all source code and unlike some other licenses, it does not require redistribution of modifications. In other words, a company can develop and add modifications to Sim42 for their own use without being forced to share those enhancements (i.e. a proprietary model of a unit operation).

The Sim42 Software Foundation [9] was incorporated to promote the development of this OSS Process Simulator. The foundation maintains the servers that provide public access to the project source code; discussion lists and other tools that may be deemed appropriate. It also decides what code and modifications are appropriate for inclusion in new versions and is the copyright holder and license grantor for all code included in the simulator.

## 6. Features of the simulator

Some of the most important and distinctive features of this only OSS process simulator are:

- The simulator is written in the Python (Brueck and Tanner, 2001) programming language, which is OSS and can be obtained at no cost via the Internet. The decision to use this programming language (Hammond and Robinson, 2000) relies on the many features offered by Python. It is object oriented, it is very high level (i.e. includes an extensive set of built-in data types and functions), has a clean syntax and it is essentially multiplatform. Another advantage of Python is that it is designed in such a way that extensions can be easily added either in Python or C++. Eric Raymond provides a thorough review of Python on his website [10].

- The building blocks of the OSS process simulator were created with well-defined interfaces thus allowing parallel and decentralized development. A clear independence from user interfaces and thermodynamic method providers is emphasized.
- The flowsheet solver can propagate partial information both backwards and forwards, which allows many complex problems to be solved without iterative calculations.
- It implements a distillation column that employs a Russell (Russell, 1983) inside/out algorithm capable of solving complex pump around, side water draws and side stripper separation configurations.
- There is no need for recycle unit operations in that estimated values are used to initialize material recycle loops.
- Balances are based on "in" and "out" ports which in turn make a "Stream" just one more unit operation. Also, connections between unit operations can be made without a stream in the middle so long as an "out" port is connected to an "in" port.
- Different thermodynamic providers can be accessed within the same flowsheet or by the same unit operation.
- Multilanguage support is provided.

## 7. Basic design of the simulator

The simulator on its own provides an interface to simulation services such as a flowsheet object, unit operations and a thermodynamic administrator. Figure 1 shows a very basic schematic of the process simulator's design. The dashed box encloses the basic objects provided in the current distribution of Sim42. Note, that the user can communicate with the Sim42 interface directly with Python code or through a Command Line Interface (CLI). The CLI is a powerful means for users to communicate with the simulator through interactive sessions or through scripts. Graphical user interfaces (GUI) have also been developed. VMGSim, which is a commercial process simulator developed by Virtual Materials Group [10] that communicates to Sim42 through the CLI. The current distribution of Sim42 also includes Simba, an interface to Sim42 that uses a web browser. An open source GUI written in wxPython [11] is currently under development.

Sim42, figure 1, does not contain a thermodynamic calculation server. It only has a Thermodynamic Administrator, which is intended to administer the communication between the process simulator and potential Thermodynamic Servers. Writing a high quality property package is no small undertaking (Agarwal, et al, 2001) but it is expected that a number of property packages, both proprietary and open source will be added. Currently, the Virtual Materials Group [1] provides a free version of their RK property package. Figure 2 presents a schematic of the possibilities for "interfacing" to Sim42. A more detailed explanation on the basic structure of Sim42 is provided in the manual [12]

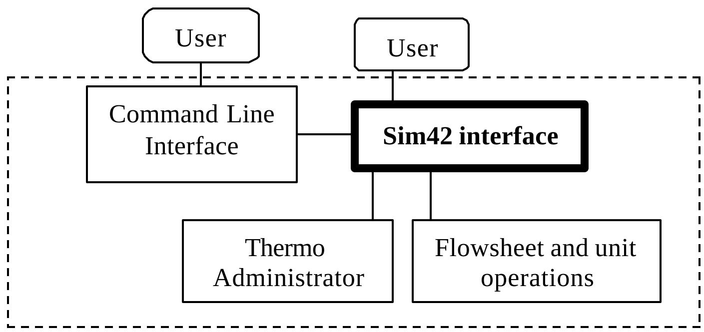



**Fig1. Basic objects of Sim42**

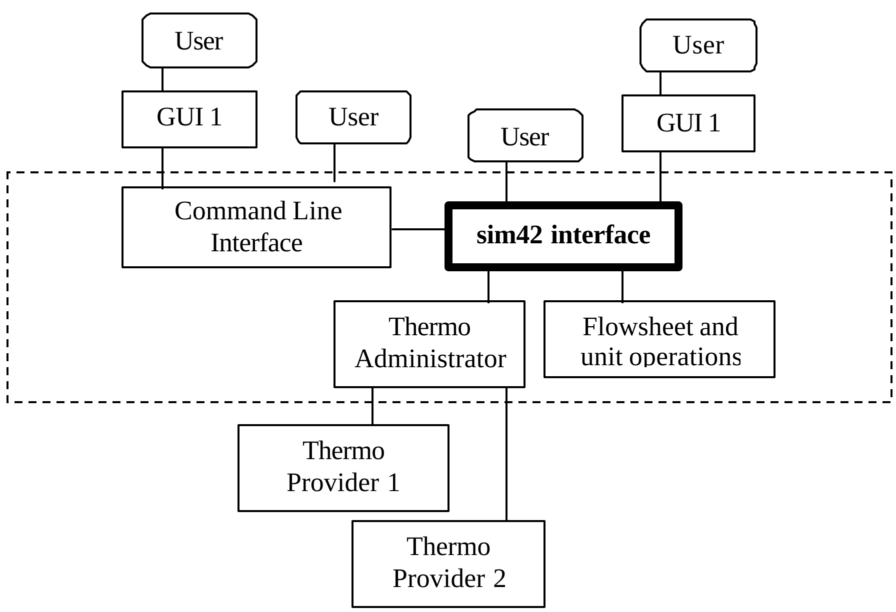



**Figure 2: Basic objects of Sim42 with custom interfaces and thermo providers**

# 8. Exploring the basic objects of Sim42

## 8.1. Unit Operations (UO)

Unit operations contain:

a) Child UOs; this allows the creation of complex UOs by using and connecting a number of simpler UOs.

b) Ports; These objects contain the property values and provide a unique means for the exchange of information between connected UOs. There are material, energy and signal ports. All these ports can be "In" or "Out" ports.

c) Parameters; In general, these are structural values that do not change during the solution of a simulation. Examples of such values are the number of UO inlet ports or the number of liquid phases calculated by the Thermo server.

d) Thermo Case; This object specifies the basic information needed to communicate with the Thermo Administrator, which in turn calls the appropriate Thermo Provider.

A “flowsheet” is a special UO that contains the algorithm for solving its child UOs. This algorithm recognizes the simulation structure, controls information propagation and recognizes the estimates provided for material loop recycle solution. Any UO can have any UO as a child, but it is important to note that a flowsheet must be on the top. Figure 3 presents a generic scheme of contained unit operations.

## 8.2. Ports

Unit operations exchange information with other unit operations by means of ports. A port is essentially an attachment point for the flow of information into and out of the unit operation. It might be a material port, which contains all of the information normally associated with a process stream (temperature, pressure, flow, composition, etc.) or it might be an energy port that just contains an energy flow or even a Signal Port that transmits a single piece of information such as a pressure drop.

A Signal Port contains a list of “Property” objects that contain such information as the name, value, conversion factor, min, max and status. “Status” indicates if the value was calculated, specified by the user, passed through a connection, etc.

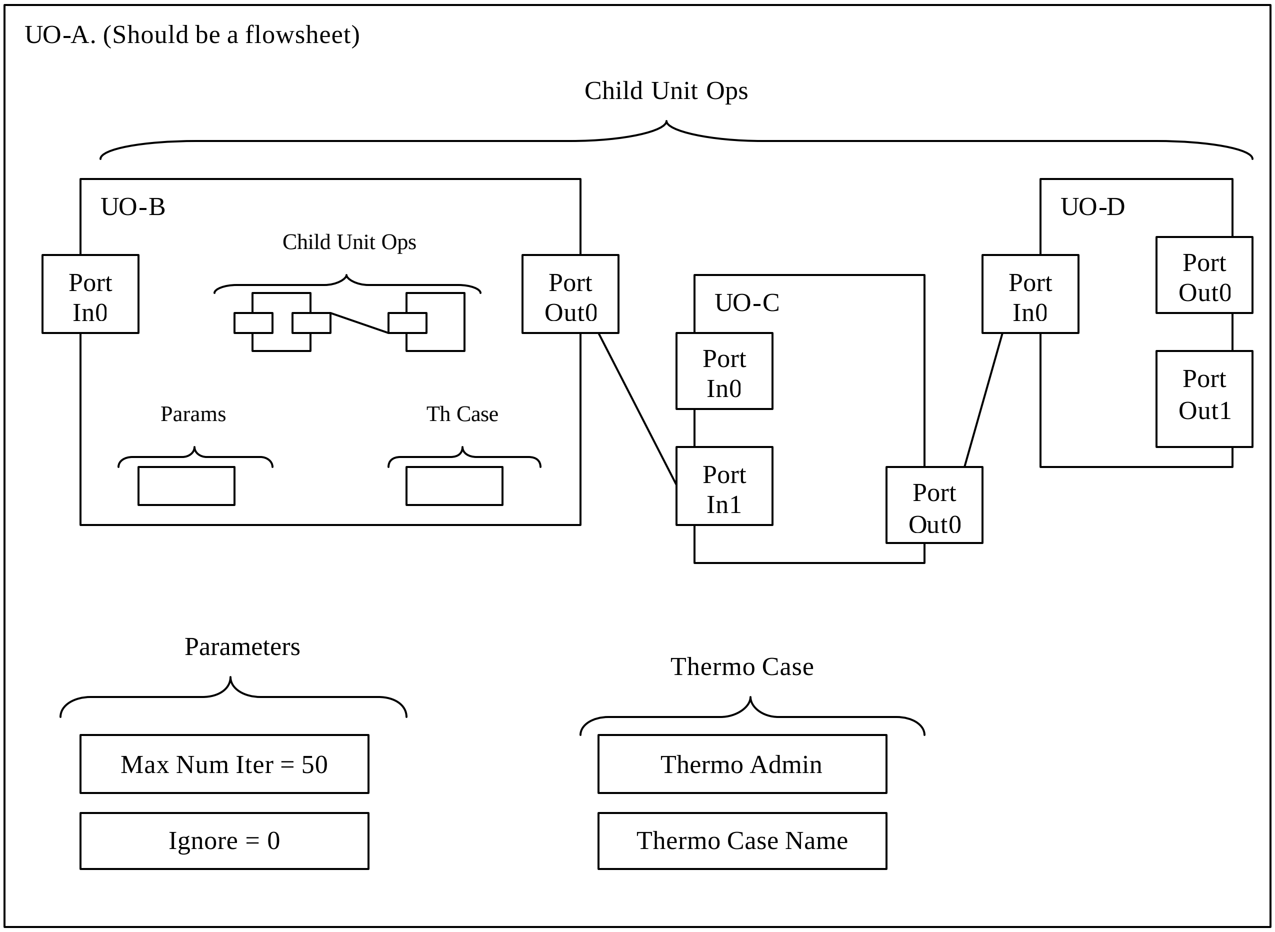



**Figure 3; A generic scheme of unit operations with a flowsheet being the parent object**

## 9. Benzene/Toluene flash calculation.

This example creates a stream and performs a flash calculation for a mixture of benzene and toluene at 80 °C and 1 atm. The simulations shown will use the interface called Simba (Sim42 Browser Assisted) which is included in the current distribution and can be run using most web browsers. Figure 4 shows a snapshot of the Simba interface. There is a multi user on-line version in the Sim42 web site which can be accessed via the following link:

**WebSite:** http://demo.sim42.org/sim42
**Login name:** guest
**Password:** sim42

Simba is a wrapper for the Command Line Interface (CLI) and processes commands entered into a the command text box located in the top section, Figure 4. A process simulation always has a "root" object (a Flowsheet unit operation) which in turn contains objects such as parameters or child unit operations. The "root" object is identified by the "/" character. The complete hierarchy of objects can be explored in the left column of Simba (Figure 4, simulation object tree) by clicking on the desired object. The main section of the application displays the "active" object and its deplay depends of the type of object being modified.

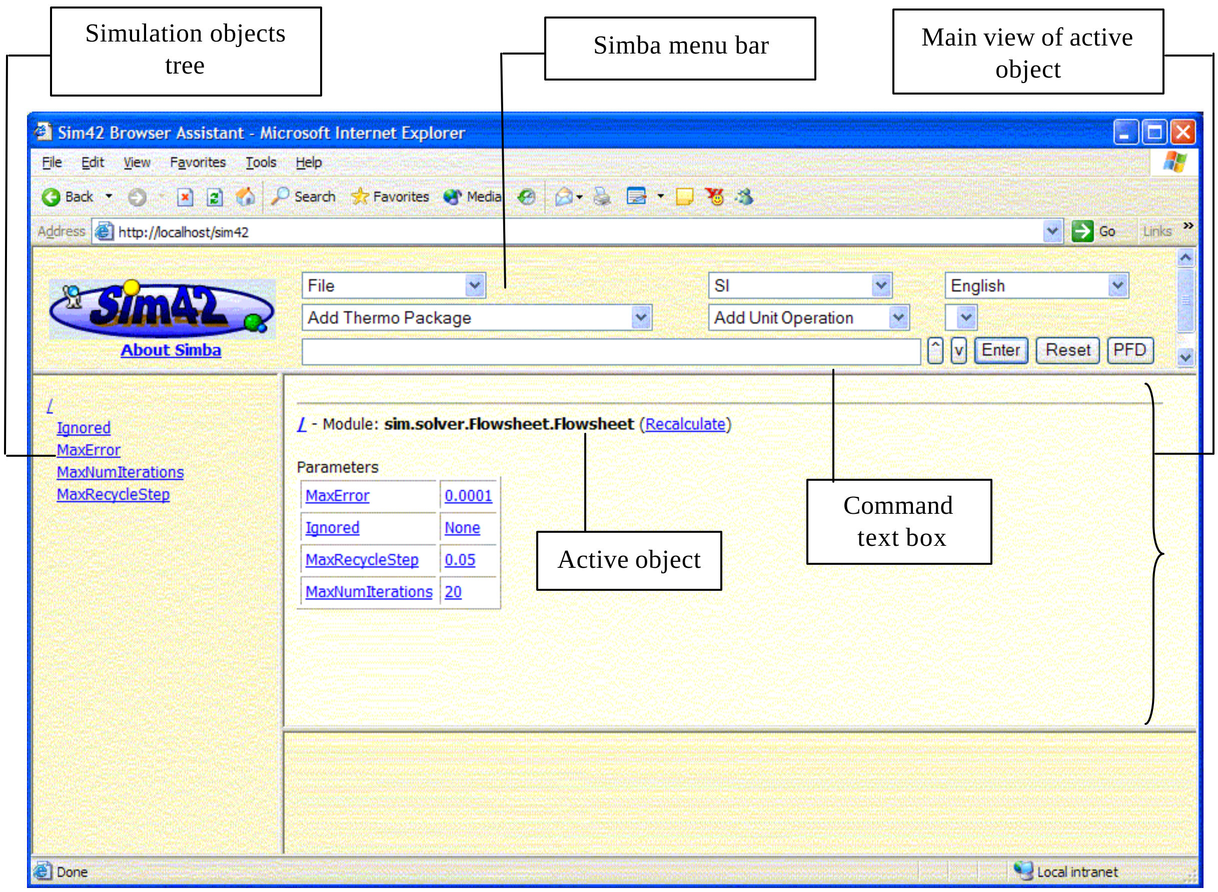



**Figure 4; Snapshot of SIMBA**

The first step in creating a simulation is the addition of a thermodynamic case, which in the simplest case involves the selection of a thermodynamic package and a set of chemical compounds. This can be achieved by selecting a property package from the combo box called "Add Thermo Package". In this case "VirtualMaterials.SRK" is selected and the name "ThermoCase" is assigned to it when prompted to do so. Note, the CLI command triggered for this action is:

*ThermoCase = VirtualMaterials.SRK*

Typing the previous line is equivalent to using the "user friendly" combo box. The following command can be typed in order to add chemical compounds to the ThermoCase (instead of selecting from the provided list). Figure 5 shows a snapshot of the added thermodynamic case.

*/ThermoCase + Benzene Toluene*

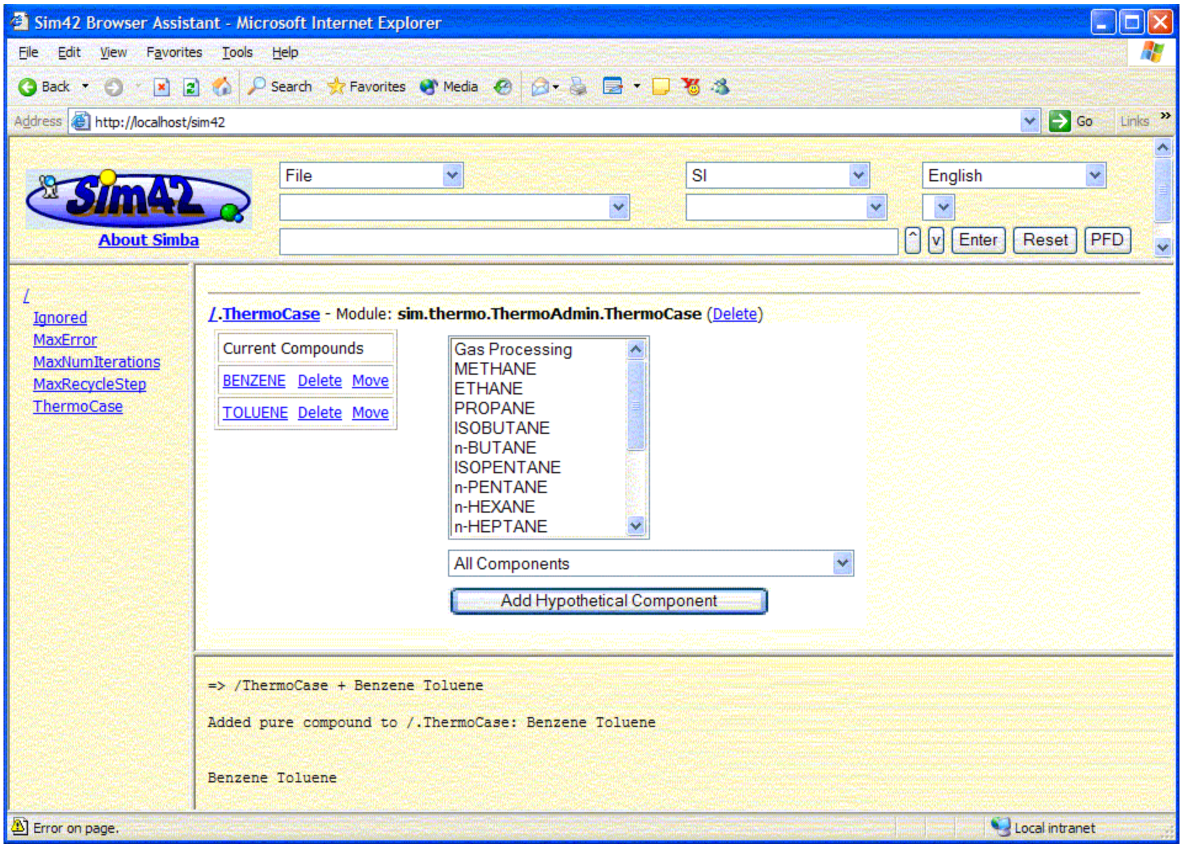



**Figure 5; Snapshot of ThermoCase in Simba.**

The next step in the simulation process is the addition of unit operations to the flowsheet. To accomplish this it is necessary to make the flowsheet the "active" object by clicking in the "/" located in the left column of Simba. Unit operations can be added by using the combo box labeled "Add Unit Operation". In this case "Material Stream" is selected and the name "s" is assigned to it. The new unit operation automatically becomes the "active" object and properties can be added to its "In" or "Out" port. The displayed units use the default set "SI" but can be changed to any available set via the corresponding combo box. For purposes of this example values are input into the "In" port (P = 100 (kPa); T = 80 (C) and Composition = 0.5 0.5 (molar)). It is important to note, that all the intensive variables of the ports for the stream appear “filled in”. Sim42 constantly monitors the degrees of freedom and performs all calculations as information becomes available. Specifying an extensive variable such as mass flow would result in a fully calculate stream. An input of 100 kg/h is used in this example and the simulation results are displayed in Figure 6.

All the previous steps could have been coded in a text file as a series of commands and then run as a script using the "Read Script" option of the “File” combo box in Simba. This same combo box provides options for storing and recalling process simulation projects, among other operations. Simba also provides an option for changing the active language to any of the currently supported languages; Spanish, French, Portuguese, Malay or English.

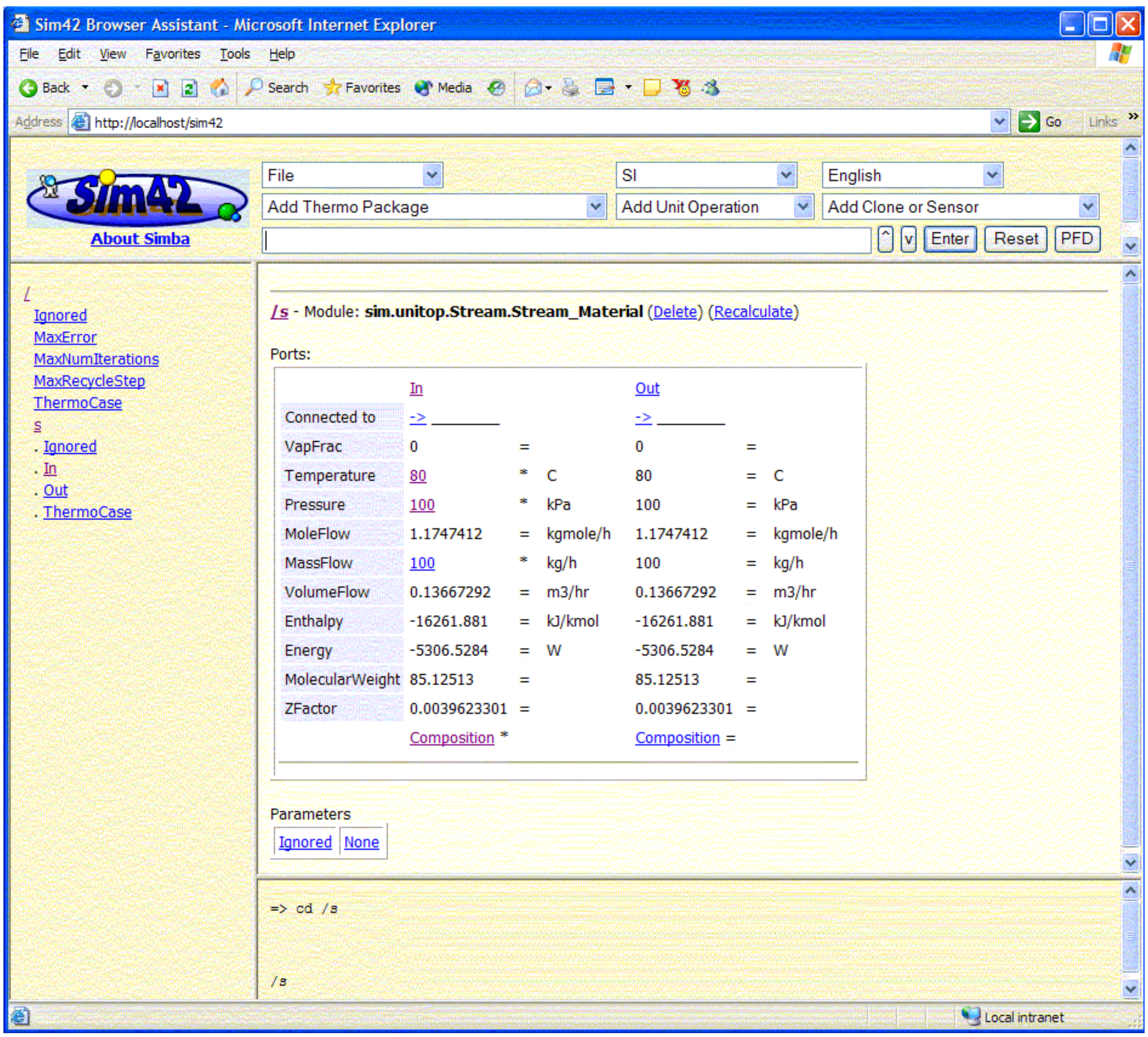



**Figure 6; Snapshot of stream “s”.**

## 10. Ethanol distillation column simulation.

This example uses live steam to fractionate an ethanol-water mixture into a near azeotrope ethanol/water at the top and a negligible concentration of ethanol in the bottoms product stream.

Again, the first step is to create the thermodynamic case, a “Feed” stream and a “Steam” stream. The values used are summarized as follows:

*thermo = VirtualMaterials.PSRK*
*thermo + Ethanol Water*

*Feed = Stream.Stream_Material()*
*Feed.In.MoleFlow = 34.43*
*Feed.In.Fraction = 0.3 0.7*
*Feed.In.VapFrac = 0.0*
*Feed.In.P = 101.325*

*Steam = Stream.Stream_Material()*
*Steam.In.P = 24.7 psia*
*Steam.In.Fraction = 0 1*
*Steam.In.MoleFlow = 51.1*
*Steam.In.VapFrac = 1.0*

In the next step a Tower is created and the name “dist” is assigned to it. The initial display of the tower is shown in figure 7. For this example, the parameter “MaxOuterLoops” is changed to 40 and twelve stages are added to the first stage. In order to add stages it is necessary to select the stage after which stages will be added. In this case “Stage_0” can be selected from the tree in the left column or the link “0” can be selected in the main view. Stages can be added using the “Add the following number of stages below this stage” text box. In this case “12” is the input.

The next step is to add a liquid draw to “Stage_0”. To do this, it is necessary to make this stage the “active” object and then select “Liquid Draw” from the “Add Feed or Draw” combo box. For this example, the name “l” is assigned to it. Adding the liquid draw makes this object the “active” object and the following values are introduced (the text written before the “>” is to indicate the active object:

*/dist.Stage_0 l.Port > P = 101.325*
*/dist.Stage_0 l.Port > MoleFlow = 12.91*

An energy draw and a tower estimate in “Stage_0” are added following the same procedure. That is: selecting “Energy Draw” from the “Add Feed or Draw” combo box and selecting “Temperature Est.” from the “Add Spec or Est” combo box. The names “cond” and “estT”, respectively are used in this example. The value for the temperature estimate is:

*/dist.Stage_0> estT = 78*

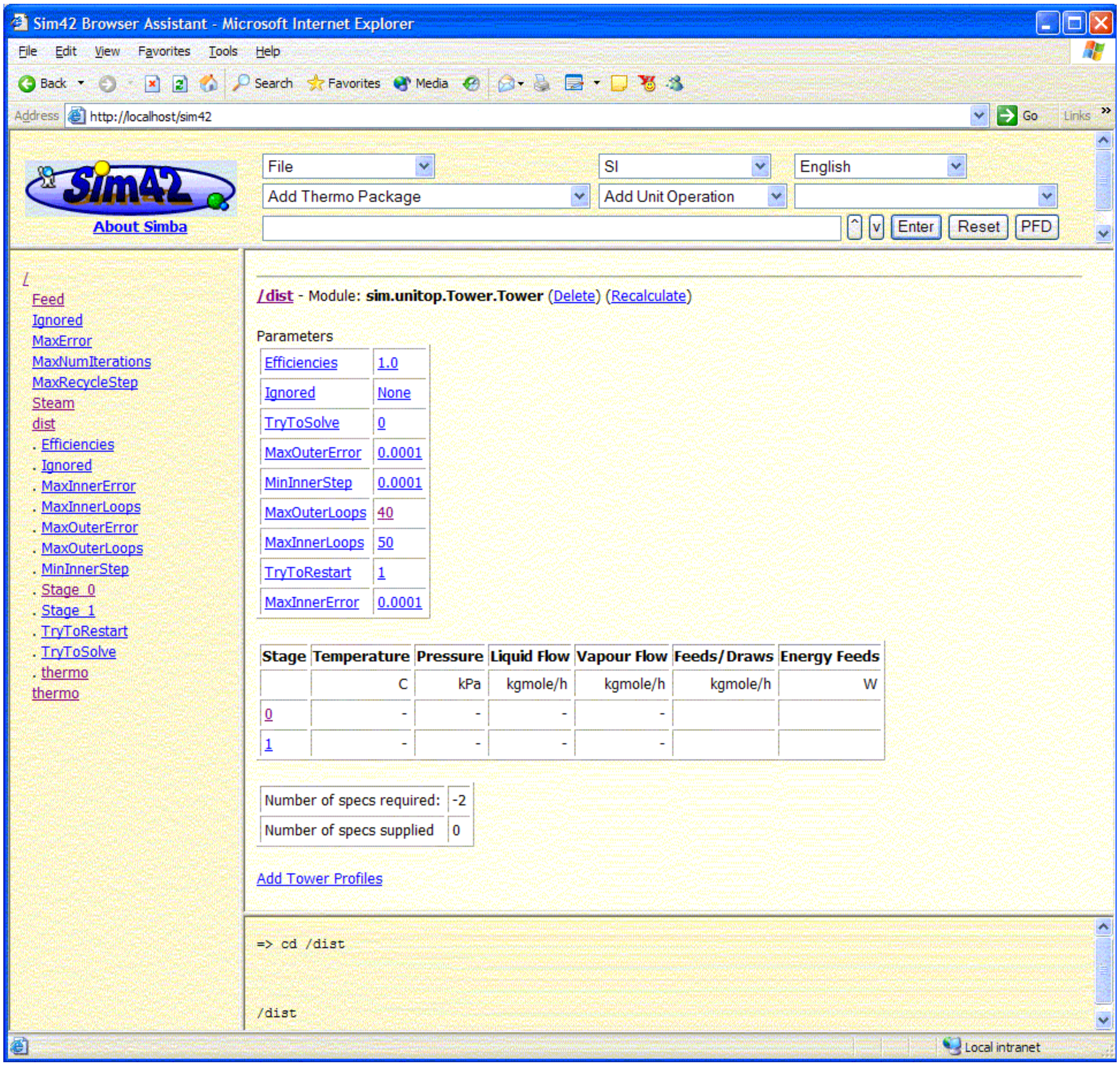



**Figure 7; Initial view of a distillation column in Simba.**

The next step is to add a feed to “Stage_11” of the tower. In order to do this it is first necessary to make “Stage_11” the “active” object. This can be done by selecting “dist” from the left column of Simba and then selecting “11” from the list of displayed stages. The feed is added by selecting “Add Feed” from the “Add Feed or Draw” combo box. The name “f” is used in this example. Once “f” is created, it can be connected to the "Out" port of the "Feed" stream by clicking in the "->" link of the "Connected to" row (assuming /dist.Stage_11.f is the "active" object). It is important to mention at this point that Sim42 has a novel port-wise design in which information to unit operations is passed by using ports. In this example, the information in the "In" port of the "Feed" stream could had been input directly into the "f" port of the "Stage_11" of the Tower without the need to create an external stream.

The only ports missing from the column are the steam inlet and the bottoms product outlet. Both ports are created at the bottom of the Tower, "Stage_13". The steam inlet port is created by adding a feed and the bottoms port is created by adding a liquid draw. Both objects are added after making "Stage_13" the "active" object and are labelled "f" and "l", respectively. The newly created feed is connected to the "Out" port of the "Steam" stream. The pressure of the bottom product liquid draw is set as follows:

*/dist.Stage_13 l.Port >P = 101.325*

Setting temperature estimates is recommended for towers in Sim42. In this example, an extra Temperature estimate is added to "Stage_13" and a value of 100 is assigned to it.

*/dist.Stage_13> estT = Tower.Estimate('T')*
*/dist.Stage_13> estT = 100*

At this point the Tower is ready to be solved and the only thing that is left to do is setting the parameter TryToSolve = 1. Figure 8 shows the tower before solving and figure 9 shows the tower after being solved. The results of the simulation are summarized in Table 1.

| Property | Feed | Steam | Distillate | Bottoms |
|---|---|---|---|---|
| **VapFrac** | 0.0 | 1.0 | 0.0 | 0.0 |
| **T (C)** | 82.30 | 115.12 | 78.35 | 99.788 |
| **P (kPa)** | 101.325 | 170.30 | 101.325 | 101.325 |
| **MoleFlow (kgmole/h)** | 34.3 | 51.1 | 12.91 | 72.62 |
| **MassFlow (kg/h)** | 910.026 | 920.56 | 521.75 | 1308.84 |
| **H (kJ/kmol)** | -26999.078 | 12894.15 | -23523.779 | -28474.91 |
| **Energy (W)** | -258216.18 | 183025.39 | -84358.88 | -574401.94 |
| **MolecularWeight** | 26.43 | 18.01 | 40.41 | 18.023 |
| **Zfactor** | 0.00134 | 0.987 | 0.002188 | 0.000832 |
| **ETHANOL** | 0.3 | 0.0 | 0.798 | 0.000289 |
| **WATER** | 0.7 | 1.0 | 0.201 | 0.9997 |

**Table 1; Results of the Ethanol production tower simulation**

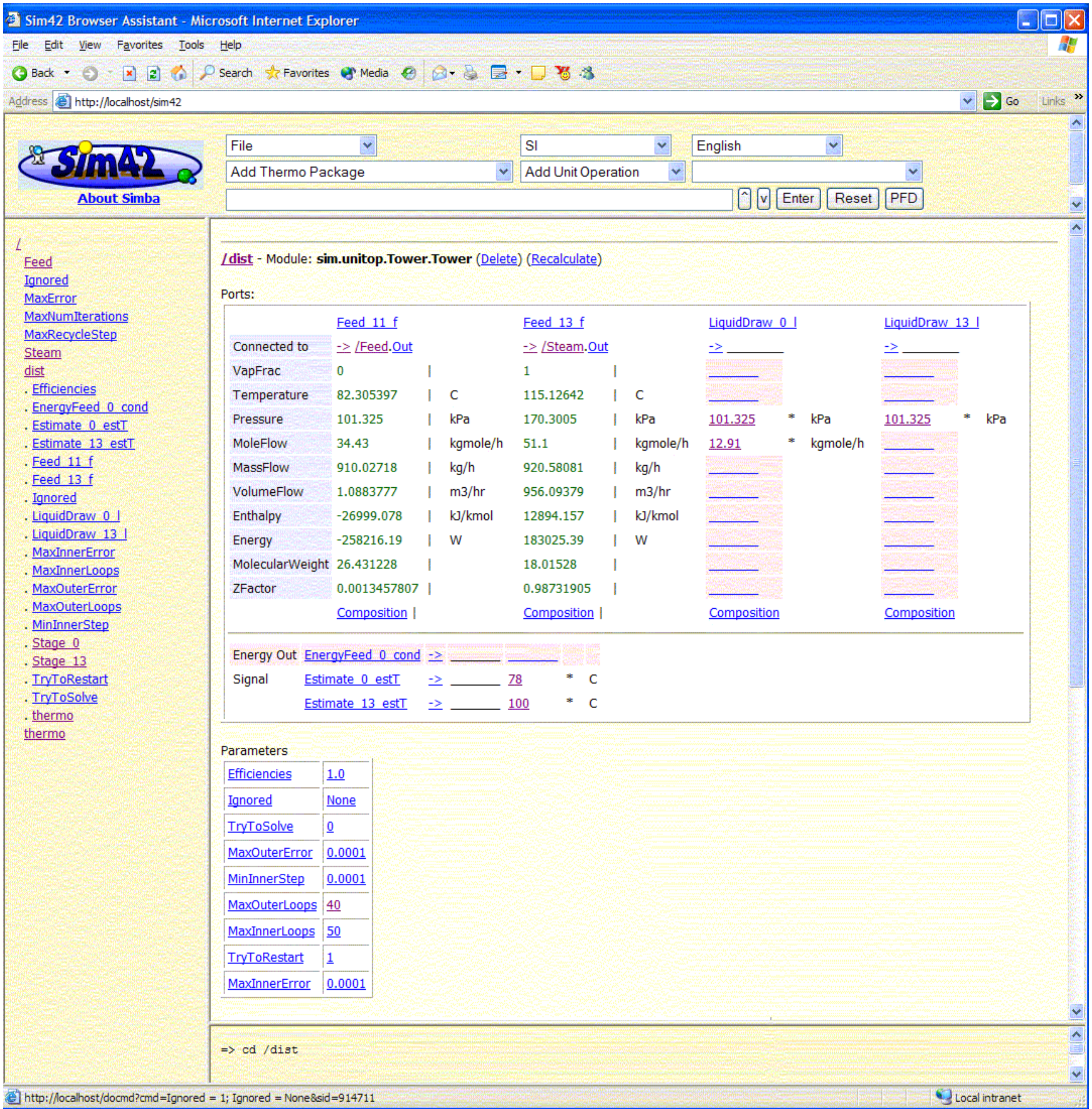



**Figure 8; Simba Tower snapshot before solving.**

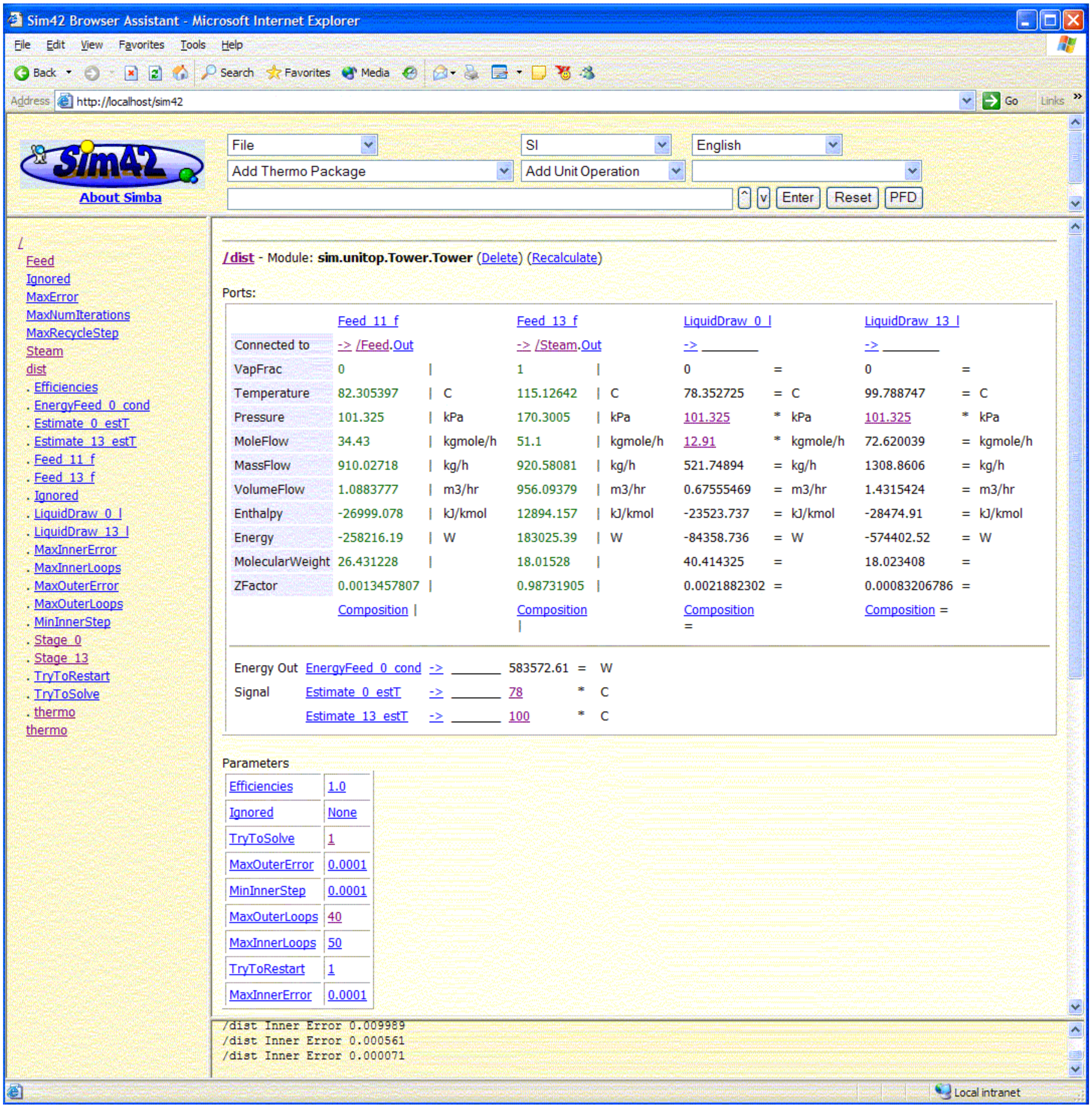



**Figure 9; Simba Tower snapshot after solving.**

The tower simulation is presented here to show the usage and creation of distillation towers in Sim42. It is important to note that Sim42 offers pre-built towers such as “Distillation Column”, “Rebolied Absorber”, “Refluxed Absorber” and “Absorber” which already contain ports for feeds and material and energy draws. Note, the pre-built towers can be modified like the tower described in the example ethanol/water distillation column simulation. Simba also provides a graphical representation of the simulations which can be accessed by clicking the "PFD" button. Figure 10 shows the Simba PFD for the ethanol/water distillation column simulation.

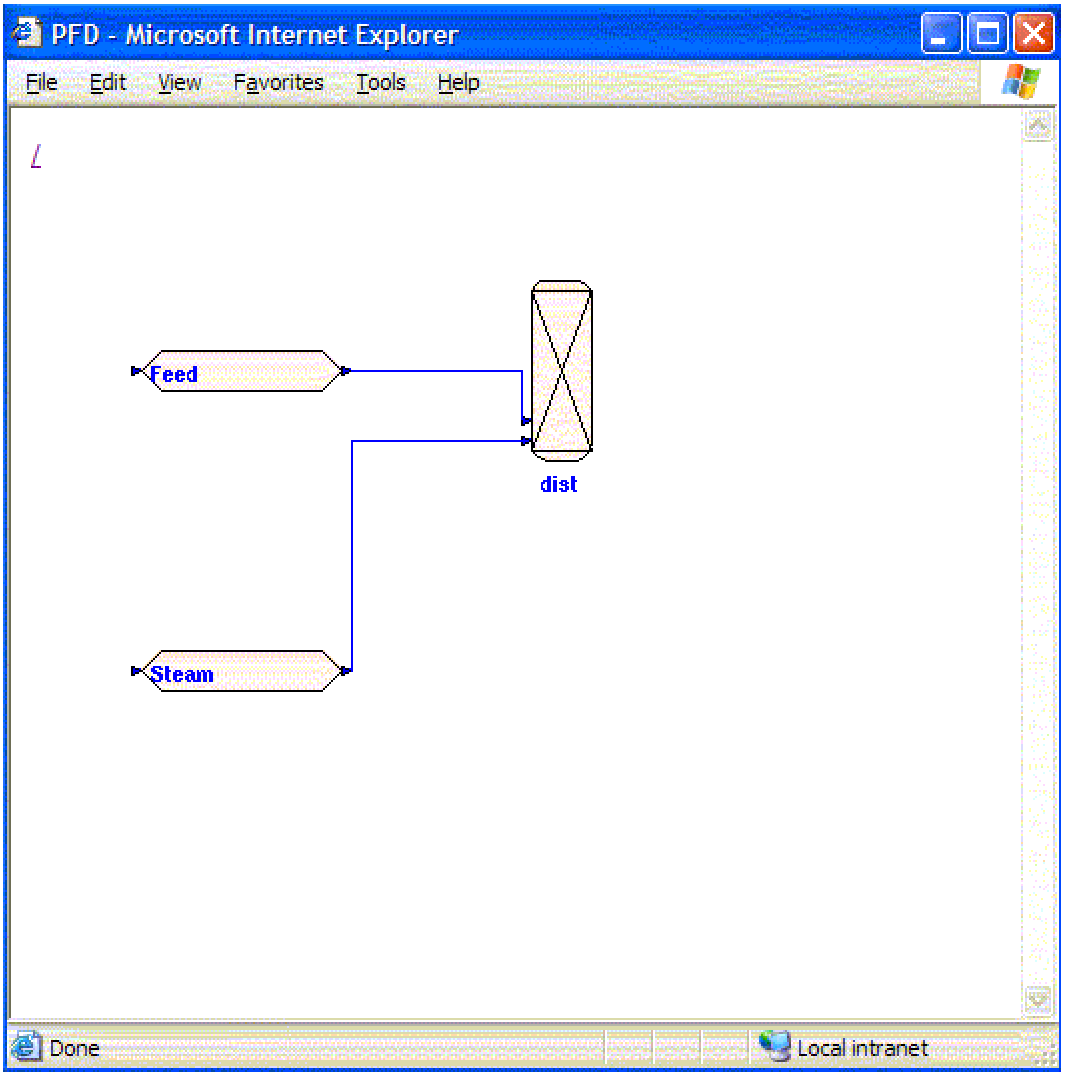



**Figure 10; Simba PFD.**

## 11. CONCLUSION

An Open Source Chemical Process Simulator was developed. It is believed that the OSS way of developing software does overcome the shortcomings posed by currently available commercial process simulators, in that Sim42 provides an inexpensive state of the art software tool for process modeling and its object oriented design and licensing allow for easy incorporation of specific developments either open source or proprietary.

The source code for the Sim42 process simulator is available at no cost and can be downloaded from the Internet. A non-profit organization called "Sim42 Software Foundation" was created to manage the project. The development base is still small and it is expected that Sim42 will naturally grow as involvement in the project grows.

The simulator, which includes a powerful rigorous distillation tower, has been used to model a variety of plants. The examples presented here were run using Simba but other interfaces are available or are being developed including a graphical interface based on wxPython and a professional commercial interface. This flexibility is only possible because the simulator core has been designed to be independent of both user interfaces and thermodynamic property providers.

## WEB SITES CITED


[1] Virtual Materials Group (VMG), http://www.virtualmaterials.com

[2] E. Raymond, The Cathedral & the Bazaar, http://www.tuxedo.org/~esr/writings/cathedral-bazaar/

[3] Full text of the Open Source Definition (OSD), http://www.opensource.org/docs/definition_plain.php

[4] Open Source Initiative (OSI), http://opensource.org

[5] OSI Licensing, http://opensource.org/licenses/

[6] Red Tree Development Inc, http://redtree.com/

[7] Concurrent Versions System (CVS), http://www.cvshome.org/

[8] BSD License, http://opensource.org/licenses/bsd-license.php

[9] The Sim42 Foundation, http://sim42.org/bylaws

[10] E. S. Raymond. Why Python?, Linux Journal (73), 2000,

http://www.linuxjournal.com/article.php?sid=3882

[11] wxPython http://www.wxpython.org

[12] Sim42 manual, http://manual.sim42.org/

## REFERENCES


- Agarwal, R., Li, Y., Santollani, O., Satyro, M. & Vieler, A., Uncovering the realities of simulation, *Chemical Engineering Progress, v97*(n5), 2001, 42-52
- Asklund, U. & Bendix, L., A study of configuration management in open source software projects, *IEE Proceedings: Software,* 2002, 40-46
- Braunschweig, B. L., Pantelides, C., Britt, H. I. & Sama, S., Open software architectures for process modeling: current status and future perspectives, *5th International conference on foundations of computer-aided process design, v100*, 1996, 220-235
- Brueck, D. & Tanner, S., Python 2.1 bible (New York, NY: Hungry Minds, 2001)
- Edwards, P. D. & Merkel, G., Impact of an open architecture environment on the design of software components for process modeling, *International conference on intelligent systems in process engineering, v92*, 1996, 307-310
- Grinthal, W., Chemputers. Process simulators shift into high gear, *Chemical Engineering (New York), v100*(n7), 1993, 153-154, 156
- Hammond, M. & Robinson, A., *Python programming in win32* (Sebastopol, CA: O'Reilly, 2000)
- Hecker, F., Setting up shop: the business of open-source software, *IEEE Software, v16*(n1), 1999, 45-51
- Moody, G., *Rebel code. inside linux and the open source revolution* (Cambridge, Massachusetts: Perseus Publishing, 2001)
- Russell, R., A flexible and reliable method solves single-tower and crude-distillation-column problems, *Chemical Engineering , v90(*n21*)*, 1983, 53-59
- Scacchi, W., Understanding the requirements for developing open source software systems, *IEE Proceedings: Software*, 2002, 24-39
- Wu, M. & Lin, Y., Open source software development: an overview, *Computers, v34*(n6), 2001, 33-38

# Getting Students to Account for Variation in their Analysis of Real ChE Processes

**Milo D. Koretsky**

**Department of Chemical Engineering**
**Oregon State University**
**Corvallis, OR 97331-2702**

**Abstract**

As educators are well aware, the customary educational setting in which students develop problem solving skills is one where the numerical values presented are specific and absolute. The deterministic nature of the end-of-chapter type problems is imbedded in their minds well before students even matriculate. However, as practicing engineers, they will confront the variation associated with measured data in the real world. A course in introductory statistics can force students to attend to the concept of variation. Statistics can be defined as the science of how to collect, analyze, interpret and present data with the purpose of understanding variation in a system. A key objective of introductory engineering statistics is to have students recognize variation is inevitable, and teach them skills to quantify the variation and make engineering decisions which account for it. The importance of statistics is well recognized in the chemical engineering community. For example, several recent articles in *Chemical Engineering Progress* have focused on applied statistics. Indeed, many chemical engineering programs have incorporated statistics into their curriculum. This paper describes efforts to infuse statistics into the curriculum at Oregon State University (OSU). The approach is primarily at two levels. A sophomore/junior level introductory statistics course, *Chemical Process Statistics*, has been developed. Concepts are introduced through case studies using industrial data, whenever possible. Statistical analysis of the data is discussed in terms of the physical process. In this way, the statistics and the science are coupled. However, these concepts are best synthesized when integrated with hands-on application of these concepts. To this end, statistical concepts are reinforced in senior lab. The content and structure of the introductory statistics course and efforts to integrate these concepts into senior lab will be discussed.

## 1. Introduction

Undergraduate chemical engineering education emphasizes *analysis* and then *design*. In the typical curriculum, the majority of the technical credit hours are devoted to fundamental science (e.g., general chemistry, physics, physical chemistry, and organic chemistry) and engineering sciences (e.g., mass and energy balances, thermodynamics, transport processes, reaction engineering, process dynamics and control). The student is then asked to *synthesize* this material in unit operations and then the capstone design course. However, the majority of graduates are hired as Process Engineers whose main focus is on *production*. Topics such as measurement system analysis (MSA), statistical process control (SPC), and design of experiments (DOE) are essential to manufacture quality products at reduced costs.[1] In fact, upon accepting their first job offer, many entry level engineers, enroll in in-house statistics related courses such as *Practical Data Analysis, Statistical Process Control,* and *Design of Experiments.*[2]

The importance of statistics is well recognized in the chemical engineering community. For example, several recent articles in *Chemical Engineering Progress* have focused on applied statistics[1,3-6]. Many chemical engineering programs have incorporated statistics into their curriculum[2]. Two ChE specific courses in applied statistics have been recently reported[7,8]. Indeed, a survey of our alumni who graduated prior to implementation of the program described below found that statistics presented the largest discrepancy between preparation at the university relative to the importance in employment[9]. Given the curricular constraints of the program, statistics in the chemical engineering department at Oregon State University (OSU) is addressed at two levels. (1) a required introductory statistics course, *Chemical Process Statistics*, is offered in the sophomore/junior year, and (2) these concepts are reinforced in the senior unit operations laboratory. To facilitate this connection, it has been found effective to have the statistics instructor give two "refresher" lectures to the lab class.

In this paper, some educational opportunities for a statistics course to address are first anecdotally illustrated with a couple of examples pulled from student work. An overview of the chemical process statistics class at OSU is then presented. This overview includes the course goals, the course learning objectives, the industrial case studies which form the heart of the class, and the assessment of the class. Finally comments are made towards the effectiveness of integration into senior lab.

**2. Educational Opportunities in Statistics from Student Work**

As educators are well aware, the customary educational setting in which students develop problem solving skills is one where the numerical values presented are specific and absolute. The deterministic nature of the end-of-chapter type problems is imbedded in their minds well before students even matriculate. However, as practicing engineers, they will confront the variation associated with measured data in the real world.

An example which illustrates this mindset follows. It comes from student analysis of a Ta etch process described in the Case Study II presented later in the paper. In Figure 1, the student presents box plots of wafer thickness vs. wafer number for 10 wafers measured, in order, from 11 different lots. These data represent the normal variation associated with a stable chemical process. Of particular interest is the curve the student drew below the box plots. The student reports,

> There appears to be a sinusoidal trend in the minimums, maximums and upper quartiles. The box plot graph shows only the minimum trend for simplicity. The sinusoid has an amplitude of 0.3 microns. While small, this might be attributed to special causes (such) as cycle contaminates.

This student is desperately looking for structure in the common cause variations associated with real processes.

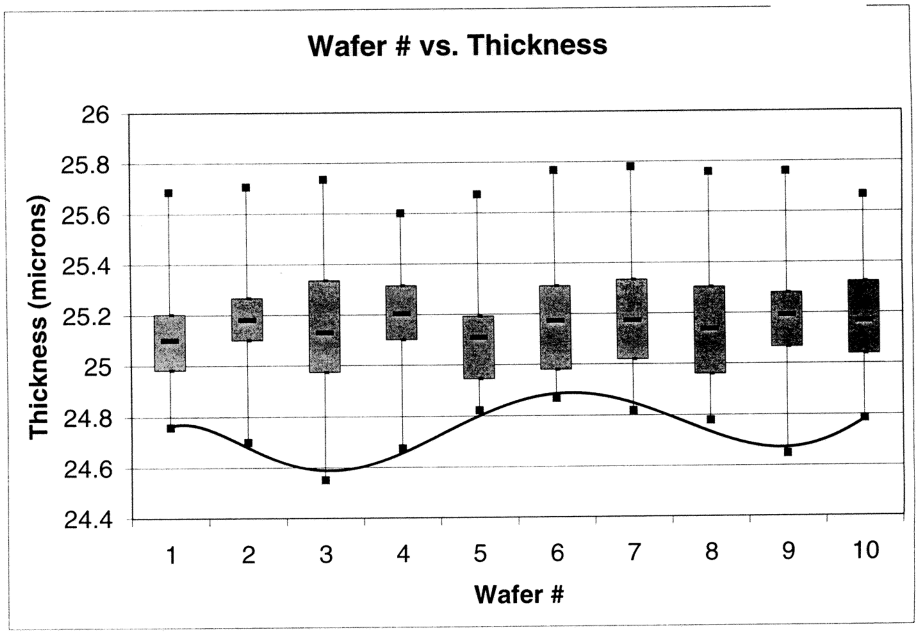



**Figure 1. Example of student looking for structure in common cause variation**

A second example comes from the heat transfer unit operation in senior lab. In this lab, the student group was tasked with comparing the overall heat transfer coefficient in two heat exchangers, the Armfield heat exchanger and the pilot scale heat exchanger, and fitting the data to a power law form. They ran the experiment in co-current and counter current configurations. The results are presented in Figure 2. When discussing the Armfield heat exchanger for co-current flow the group reports, " with an $R^2$ = 07566. Statistically, this is a poor fit for the data..." On the other hand, for co-current operation in the pilot scale they report, "$R^2$ = 0.9398. Statistically this is an acceptable correlation." This group is blindly applying the value of the correlation coefficient to draw a conclusion without realizing it is much easier to fit four data points to a power law correlation than seventeen. The logical conclusion that they would draw is that the correlation given by the four points for the pilot heat exchanger presented on the right of Figure 2 are statistically more reliable than the fit for the Armfield heat exchange on the left!

These examples are but two of many; you may even have your own stories. They are provided to illustrate the conceptual areas in designing an introductory statistics course and integrating statistics into the curriculum. The first example shows that students need to conceptually recognize the variation in real measurements. This task is challenging in light of the deterministic nature that science and engineering is typically taught. The second example illustrates two points. First, the statistical methodology should be understood well enough that proper interpretation is given to the statistics used, such as to $R^2$ in that example. There is a second, subtler, lesson as well. In the context of the second example, after learning statistics, one would hope the students can recognize to ask, "Is there any difference between the co-current and counter current configurations in the Armfield heat exchanger?" Moreover, to realize that statistics can be used to answer such a question and that it may be more appropriate to fit one expression to all the data rather than separate expressions for co-current and counter current configurations.

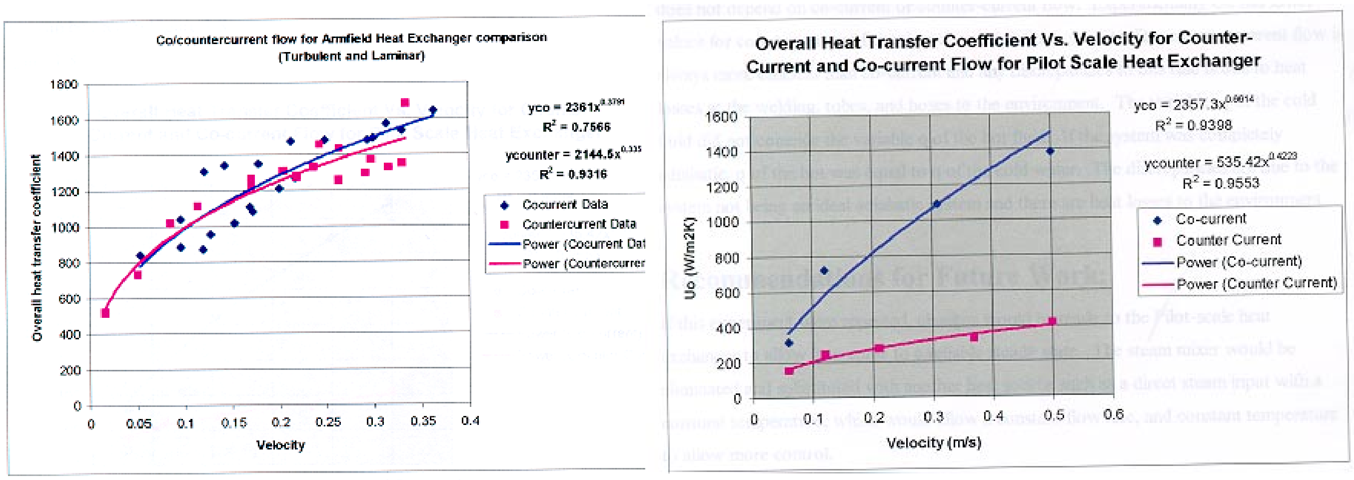



**Figure 2. Correlation of heat transfer coefficients in senior lab.**

## 3. ChE 302: Chemical Process Statistics

### 3.1 Course overview

*Chemical Process Statistics* (ChE 302) was developed to provide students exposure to statistics, in the context of the educational challenges discussed in Section 2. It especially focuses on those topics that will be useful for work in industry. The Course Goals and Course Learning Objectives are presented in Figures 3 and 4, respectively. An outline of the topics is presented in Table 1.

The course content reflects, for the most part, topics covered in many engineering statistics textbooks[10-19]. However, one topic not commonly covered is measurement system analysis. measurement system analysis evaluates the instruments used to measure a process in order to determine their accuracy and estimate the sources of variation and their extent. The process of evaluating a particular set of measurement instruments is often called a gauge study. For example, a gauge study is the first step in the formal process and equipment qualification plan developed by SEMATECH, a government supported consortium of major US semiconductor manufacturers[20,21]. In fact, many interns and recent college graduates are tasked with executing

**<u>Course Goals</u>:**

1. Develop an awareness of the utility of statistics in assessing experimental data and operating industrial chemical processes.
2. Describe the basic concepts and nomenclature associated with applied statistics, Measurement System Analysis, Statistical Process Control, and Design of Experiments.
3. Work through real industrial examples (case studies) in the field of chemical engineering to gain experience with these tools.
4. Utilize computer software (Microsoft Excel, StatGraphics) to aid in statistical analysis.

**Figure 3. Course Goals for ChE 302, *Chemical Process Statistics***

**Course Learning Objectives**

*By the end of the course, you will be able to:*

1. Define major terms used in applied statistics including those on the assessment matrix.
2. Define the typical steps in analysis of data. Apply these steps when you treat measured data.
3. Calculate measures of central tendency and dispersion based on data and frequency distributions, and make the following plots: Box Plot, histogram, run chart.
4. Define variation. Identify different factors which contribute to variation. Estimate whether a source of variation is due to a common cause or a special cause.
5. List the major factors that affect measurement system analysis. Calculate the repeatability and reproducibility of a gauge based on measured data. Calculate the precision to tolerance ratio.
6. Use sampling distributions to estimate confidence intervals based on measured data.
7. Set up a hypothesis test to statistically make a decision.
8. Define when a process is statistically in control. Given data from a process, calculate control limits and capability ($C_P$ and $C_{pk}$).
9. Perform least squares regression analysis to fit experimental data to an empirical model equation. Calculate the value for the regression coefficient.
10. Quantify the effect of (i) a single factor and (ii) two factors on a process by applying Analysis of Variance (ANOVA).
11. In the context of Design of Experiments (DOE), (i) set up a balanced design array, (ii) create a marginal means plots and/or an interaction plot from the experimental response, and (iii) develop an empirical model equation.

**Figure 4. Course Learning Objectives for ChE 302, *Chemical Process Statistics***

gauge studies. This topic not only gives process engineers a useful tool for immediate practice but also provides a useful platform to learn about variation and variance. A more detailed description of this topic is presented elsewhere[22]. The fundamental concepts are introduced in class and reinforced in homework as in a standard lecture class. However, intertwined with this presentation are case studies using industrial data where statistical analysis of the data is discussed in terms of the physical process. In this way, the statistics and the science are coupled. The case studies covered in ChE 302 are described briefly below.

Throughout the term, two software programs are used to perform calculations on larger data sets in homework or case studies. A standard spreadsheet program, Excel, is used since it is readily available. Additionally, StatGraphics is used as an example of a statistics specific software application. StatGraphics is the university licensed software available to students. There are many common statistics packages used including: Minitab, Statistical Applications System — JMP, and Statistica. However, the objective is not to train students on a specific software package, but rather so that they get exposure to the general structure of a computer software packages. Two computer labs provide hands-on introduction to these programs.

**Table 1. Chemical Process Statistics: Course Outline**

| Topic |
|---|
| 1. Typical steps in analysis of data: AIChE Salary Survey |
| 2. Measurements of central tendencies and dispersion |
| 3. Graphical treatment of data (e.g. Box Plots and Pareto diagrams) |
| 4. Probability |
| 5. The normal distribution and other probability distributions |
| 6. Variation: common causes and special causes |
| 7. Measurement System Analysis |
| 8. Sematech Qualification Plan |
| **Case Study**: Gauge capability of video micrometer |
| 9. Sampling from populations - Student t distribution |
| **Case Study**: HP Ta etch linewidth batch process |
| 10. Confidence Intervals and Hypothesis Testing |
| **Case Study**: Ta etch batch vs. continuous processes |
| 11. Statistical Process Control |
| **Case Study**: Control of Cu Etching |
| **Case Study**: CD control in photolithography |
| **Case Study**: Ta batch etch control limits |
| 12. Curve Fitting — linear regression |
| 13. Analysis of Variance - ANOVA |
| 14. Design of Experiments |
| **Case study**: Design to improve uniformity in plasma etch |

## 3.2 Industrial Case Studies

The heart of *Chemical Process Statistics* is applying the concepts listed in Table 1 to real manufacturing data from chemical processes. Statistical analysis of the data is discussed in terms of the physical process. In this way, the statistics and the science are coupled. Moreover, it allows students to experience how these concepts often need to be extended when applied to the complexities in a manufacturing environment. Most of the case studies are taken from the microelectronics industry since this is the instructor's area of expertise; moreover, the majority of OSU BS ChEs have been placed in this industry. However, the principles can be applies to any chemical process. The case studies will be briefly described below. For more details, see references 22-25.

### Case Study I: Measurement system analysis (Gauge R&R Study)

Experimental or process data are obtained through a measurement system. Values of variables such as temperature, pressure, flow rate, concentration, thickness, etc. are needed to analyze and control processes. If we are not able to adequately make measurements, we cannot hope to make useful decisions. The first step in assessing and analyzing data should be to characterize the measurement system through a gauge study. The gauge study introduced in *Chemical Process Statistics* is based on data collected by an OSU ChE interning at Merix Corporation, a printed circuit board manufacturer[a]. This study was performed to evaluate the capability of a

[a] While the data from Merix were used, the analysis used in class differs significantly.

video micrometer in use and to assess if newer instruments needed to be purchased. While the experimental design is an important component to this process, that methodology is covered later in the course and the design that was used is simply presented.

In this analysis, the class examines components of variation due to repeatability and reproducibility. The repeatability measures the variability inherent in the micrometer, itself, while the reproducibility in this gauge study is characterized by the variation in values between operators. Reproducibility can also be assigned to different environments, different gauges, etc. From these values, a precision to tolerance ratio is determined. Calculations are performed using both nested and non-nested designs, concluding the nested design is better. The students then repeat the analysis using StatGraphics. They discover that the program's default is a non-nested design. They are then shown the less straight-forward way to do the calculation for a nested design with StatGraphics. In additional to learning the details of calculating the precision of a measurement gauge, this case study introduces the class to a methodical account of different sources of variation and shows them the pitfalls of blindly using computer programs for analysis.

**Case Study II: Comparing variation in two processes: batch vs. semi-continuous Tantalum Etch process**

Data from two process alternatives in the manufacture of ink jet printer pen heads at Hewlett-Packard Corporation are compared. A schematic of these process alternatives is shown in Figure 1. The data represent the measurement of the width (µm) of a portion of thin-film tantalum that is defined through a wet chemical etch. The original process (Figure 1a) is a batch process in which 24 wafers (1 lot) are all etched at once in a bath of etchant. The new process (Figure 1b) is a continuous process in which each wafer is processed individually in a tool that is constantly bleeding old etchant solution while gaining fresh solution. There are six processing chambers in the tool, so the first wafer goes into chamber 1, the second to chamber 2 etc. It is not unreasonable to assume that all six chambers do not perform exactly alike, so the tool can be thought of as really six separate tools. The data was taken as follows: 11 lots of wafers were measured, 10 wafers measured from each lot, 5 "sites" were measured on each wafer.

Several studies were performed with these data. Students were first asked to examine data from the batch process. They performed summary statistics, such as lot to lot, wafer to wafer and site to site means and standard deviation, and to put the data in graphical form. From this analysis, they were asked to identify opportunities for improvement. Next, students are asked to determine if the continuous process represents an improvement over the batch process. The box plots presented in Figure 1 represent data from the continuous Ta etch process. To accomplish this task, they needed to identify that variation reduction is the primary issue, since they could easily change the centering of the distribution as needed. Thus, they needed to identify, and compare the Chi-Square distribution for each process. In Case Study V they are asked to construct control charts from these data, as will be discussed later.

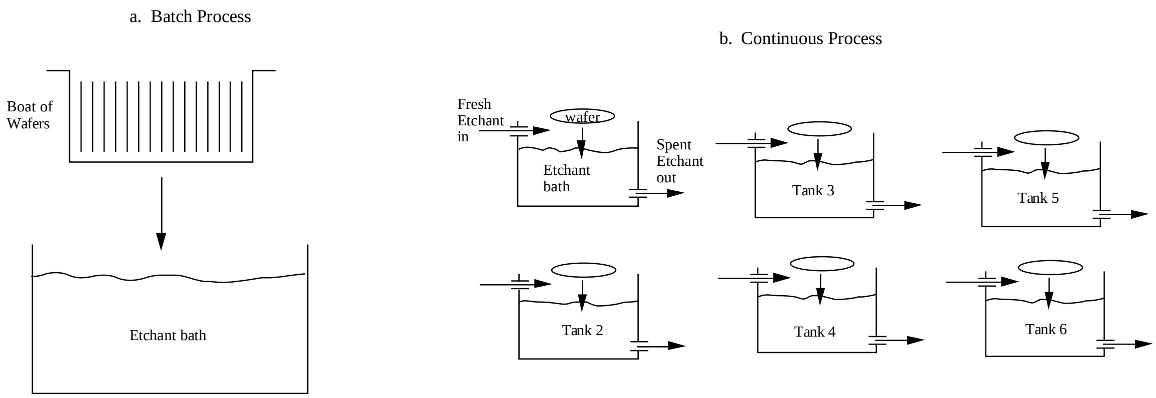



**Figure 5. Comparison of the a) batch and b) continuous Ta etch processes from Hewlett-Packard**

## Case Study III: Statistical Process Control: Copper Etching in Printed Circuit Board Manufacturing

A principle unit process in the manufacture of printed circuit boards is the patterned copper etch. This process defines the "wiring" of the electronic circuit which connects the components which are mounted on the board. Copper is etched by the following disproportionation reaction in which cupric ion ($Cu^{2+}$) reacts with metallic copper to form cuprous ion ($Cu^{+}$):

$$Cu + Cu^{2+} \rightarrow 2Cu^{+} \qquad (1)$$

Etchant is then regenerated as follows:

$$Cu^{+} + HCl + \tfrac{1}{2}H_2O_2 \rightarrow Cu^{2+} + Cl^{-} + H_2O \qquad (2)$$

As time goes on:

1. $Cu^{2+}$ accumulates in the bath as the copper clad is etched.
2. $HCl$ is consumed by reaction 2
3. $H_2O_2$ is consumed by reaction 2

To control this process we must

1. Determine how much $H_2O$ to add to dilute the cupric ion accumulation.
2. Determine how much $HCl$ to add to account for how much is diluted by control 1 above and by consumption in reaction 2.
3. Determine how much $H_2O_2$ to add to account for how much is diluted by control 1 above and by consumption in reaction 2.

Attention was focused on items 1 and 2. One particular educational benefit of this system is the intricacy of the control. It is desired to measure and respond to the concentrations of $Cu^{2+}$ and $HCl$ . However, they cannot be measured on the production line. Therefore, specific gravity and conductivity are tracked and correlated to concentrations via laboratory quantitative analysis. When the conductivity falls below a "low" setpoint, concentrated hydrochloric acid is added to the chemical sumps until the conductivity reaches an upper setpoint. Specific gravity is monitored by the position of hydrometer float in relation to an inductive sensor. When the density of the solution increases, the hydrometer rises. When the float rises above the sensor position, the inductive contact opens. Water is added until the density decreases and the float re-makes the sensor contact. The conductivity controller setpoints are numerically entered in the PLC program and can be changed. The density controller sensor position can be adjusted. This system also includes an adjustable "stop" which prevents the float from falling below the sensor. If the stop were not there, a decrease in density would cause the float to drop, opening the contact and adding water, thus further decreasing density.

One of the problems with this control system is interactions between the measured variables, i.e., their relation to the chemical constituents. Adding concentrated *HCl* not only increases conductivity, but also specific gravity. $Cu^{2+}$ also does the same. It is not hard to imagine that a control system can be configured that adds *HCl* when the density is to low and adds water when the conductivity is too high will also work in this application (and is, in fact, used in other systems). Hence the system can take a long time to stabilize. An adjustment for $Cu^{2+}$ will impact *HCl* and vice versa. The system can also stabilize at three different points. For instance, at a particular setting, one can have "good" $Cu^{2+}$ and *HCl* readings or "high" $Cu^{2+}$ and "low" *HCl* or "low" $Cu^{2+}$ and "high" *HCl*. Lab analysis is used to avoid these situations. Students are given sets of lab data of $Cu^{2+}$ and *HCl* concentrations. They are asked to determine control limits and identify special causes. They then calculate the process capability indices $C_p$ and $C_{pk}$. They discover that there exists a high number of special causes, so criteria must be applied judiciously. They also can compare the line during start-up and when it is more mature. The sluggishness of stabilization is also demonstrated.

**Case Study IV: Statistical Process Control: CD control in photolithography**

This case study involves implementation of SPC at Digital Semiconductor[24]. The process examined is during photolithography and etch in integrated circuit manufacturing where wafers are batch produced in lots of 25. After etch, the width of the remaining line is a very important parameter in device operation and is called the critical dimension, CD. Students are provided with a run charts of the average and range of CD for a set a batches. The problem with implementing SPC in this case is that an individual die on a wafer is more likely to be defective if its neighbors are defective.

Students are asked to calculate the control limits for the photolithography data and plot control lines on the run charts. The limits they calculate when they use algorithms learned in class are shown in the top of Figure 6. They identify six points beyond the control limits, and many others just within the limits. This number is unusually high for a stable process. The possible causes for this result are discussed in class[b]. In the discussion, the *assumption* of random and independent samples is scrutinized. An alternative solution is presented, where the control limits are calculated based on an *effective sample size*, $n_{eff}$, as follows. Instead of assuming that the site readings form n independent, random samples, the question is posed in reverse. If the data were hypothetically selected from this process, with mean $\bar{x}$ and standard deviation, s, how many independent readings would it represent? This approach leads to control limits plotted on the

[b] In fact, the failure of this algorithmic approach led to resistance at Digital towards implementing SPC, and motivated the solution reported in this case study.

bottom of the figure. The process appears under control, with only the run in batch 40 showing a special cause.

In the homework assigned following this case study, students are asked to form control charts from the batch process in the Case Study II. If they use the same principle as the Digital study, they get reasonable limits for this batch process.

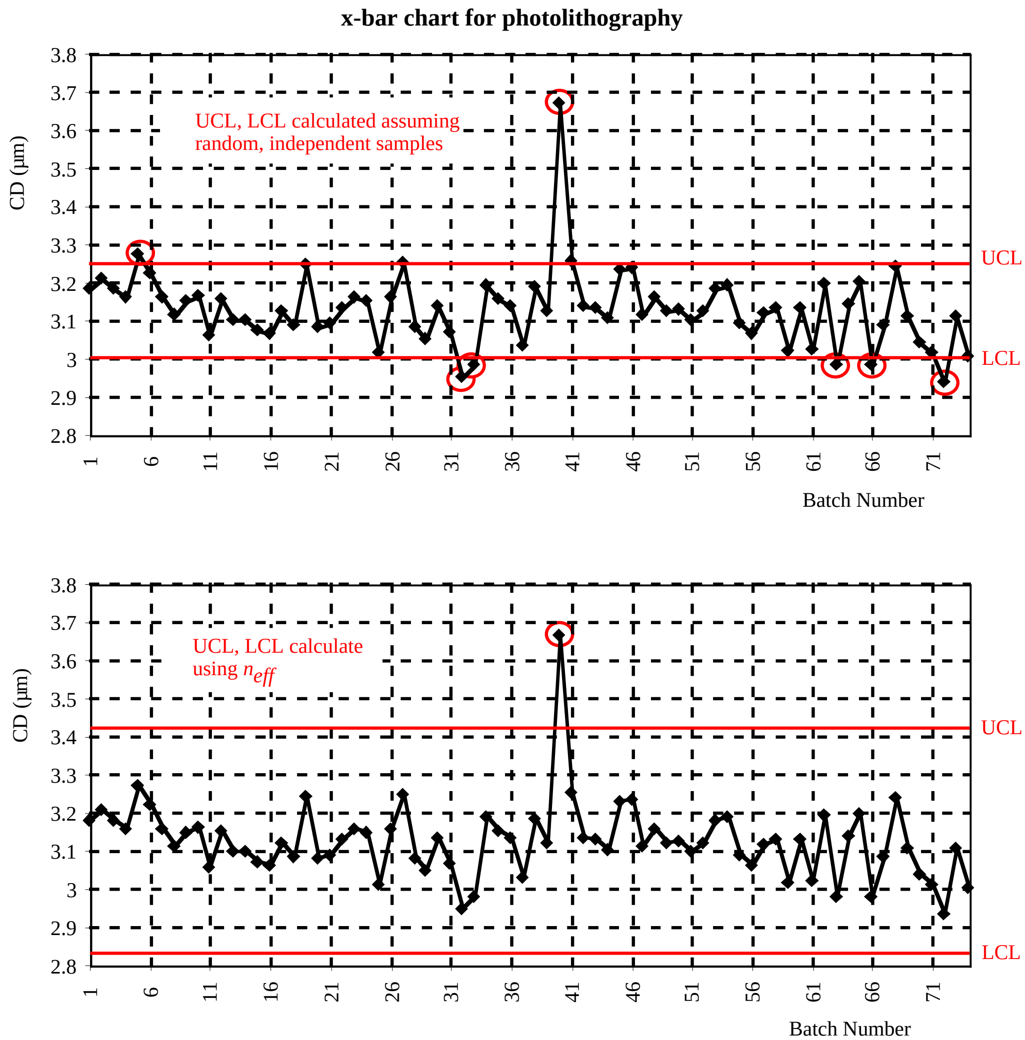



**Figure 6. Upper control limit (UCL) and Lower control limit (LCL) of critical dimension after etch. The top plot shows limits calculated for random, independent samples while the bottom plot uses $n_{eff}$. Data modified from (24).**

## Case Study V: Design of Experiments: Uniformity in oxide etching

The final case study is presented after the class has looked at some straight-forward, textbook examples of design of experiments. It is the most complex study of the term and is intended to have the seminar-like quality of illustrating and extending the concepts they have been learning. This study looks at complexities in the design of experiments that arise from confounding of the interactions. The goal of this study was to improve the uniformity obtained in etching of $SiO_2$. It was conducted at SEMATECH[25].

First a six factor two-level design using 16 runs is analyzed. This original fractional factorial design leads to 3 significant factors and nine interactions. Thus, not enough information is available to resolve the confounding. Instead of performing an additional 16 runs, semifolding on one factor allows most of the interactions to be de-convoluted in eight runs. Thus, experimental time is reduced. Physical arguments allow the remaining confounded interactions to be determined. A confirmation run shows greater uniformity and validates the analysis. This example pushes the understanding of the class. However, in this way, it appears a large group are imparted with an understanding of interactions and design arrays on a deeper level.

### 3.3 Course Assessment

Assessment of *Chemical Process Statistics,* ChE 302, was performed to evaluate if the learning objectives shown in Figure 4 were achieved by the students. The assessment period covers three course offerings from 1999-2002. Feedback has been obtained from the ChE department's industrial advisory board, from student interns and from a multi-faceted in class assessment process.

The chair of the industrial advisory board wrote:

> Degreed chemical engineers are an important resource for our company as well as other related microelectronics companies. My historical experience in interviewing graduates of chemical engineering departments from a number of institutions is that there seems to be a lack of good understanding of the concepts which are addressed in *Chemical Process Statistics*. In short, this course is meeting our needs.

In the department sponsored internship program, MECOP, students give an oral presentation after each of their two, six-month industrial internships. They are asked to list the classes they found particularly useful on their internship. *Chemical Process Statistics* was by far the most common class cited, appearing in 78% of the student presentations over the last three years. The next highest class was cited, *Technical Writing*, appeared in 46%. These results should be interpreted with caution since students have not yet taken some ChE core classes before their internships. Moreover, other classes may develop an overall thinking processes rather than specific skills, and their value may not be so readily apparent to the student. None the less, clearly ChE 302 is of value.

The in-class assessment in 1999 differed from the later two years as the process was modified to conform to the departmental approach to ABET 2000. In the first year, the students qualitatively assessed the coverage and comprehension of the seven learning objectives at the time. The results, modified to correspond to those objectives presented in Figure 4, are shown in Table 2. Exam questions also were recorded according to their corresponding learning objective. The average percentage that the class obtained as well as the standard deviation is also reported in Table 2. There is good correlation between the Exam results and the students' self-assessment. This process indicates, overall, the students have achieved the stated learning objectives. The topic picked to improve in Fall 2000 was ANOVA analysis.

**Table 2: Assessment Summary from Fall 1999**

| Learning Objective # | Exam Scores Midterm or Final | Student Assessment Coverage | Comprehension |
|---|---|---|---|
| 1 | 85–17% | | excellent |
| 3 | 68–21% | good | excellent |
| 5 | Case study | good | good |
| 7 | 68–20% | may need more | good |
| 8 | 76–19% | good | excellent |
| 10 | 60–28% | may need more | good |
| 11 | 72–26% | may need more | good |

In Fall 2000 and Spring 2002, the following assessment tools were used:

(i) **Homework Scores (HW)**. There were eight homework assignments (approximately 30 problems total) assigned in ChE 302. Each problem was recorded according to its corresponding learning objective. The average percentage that the class obtained as well as the standard deviation is reported in Table 3.

(ii) **Midterm and Final Exam Scores**. Exam questions were recorded according to their corresponding learning objective. The average (avg) percentage that the class obtained as well as the standard deviation (stdev) is reported in Table 3.

(iii) **Lab Scores** Two in class computer based lab exercises were conducted. The average percentage that the class obtained as well as the standard deviation is reported in Table 3.

(iv) **Learning Objective Self Assessment (Self).** The survey was administered on the last day of class as part of course evaluation. The percentage mastery perceived by students along with the standard deviation is presented in Table I (4.0 being 100%).

(v) **The Language of Statistics Assessment.** The survey and the results for a sample year, 2002, are shown in the Appendix. This survey was administered both before and after the course was taught. Before the course the average of all words during the three years was **2.14** with a range of 0.11, indicating the students' belief: "I have heard this word used, but I am not sure what it means " After the course, the average was **3.66**, with a range of 0.18 indicating "I can define this word."

The homework average vs. learning objective varied from 76% to 96%. The exam average varied from 59% to 94%. The lab averages varied from 93% to 95% and the self evaluation from 66% to 88%. As indicated from the table above as well as the language of assessment survey, learning objectives 7-9 appear to be lower than 1-6. This results from several possible factors - (i) more difficult material, (ii) learning in progress when surveys were conducted, (iii) not enough coverage of material. More effort will be made next year to go through LO 1-6 more quickly to leave more time for 7 and 8.

**Table 3: Measurement of Learning Objectives Fall 2000 and Spring 2003**

| LO | HW | | Exams | | Labs | | Self | |
|---|---|---|---|---|---|---|---|---|
| | avg | stdev | avg | stdev | avg | stdev | avg | stdev |
| 1 | | | 86% | 10% | | | 77% | 24% |
| 2 | | | | | | | 83% | 14% |
| 3 | 84% | 13% | 65% | 32% | | | 88% | 11% |
| 4 | | | 94% | 3% | | | 83% | 18% |
| 5 | 88% | 10% | 67% | 22% | 93% | 6% | 73% | 19% |
| 6 | 87% | 12% | | | | | 81% | 21% |
| 7 | 82% | 14% | | | | | 79% | 27% |
| 8 | 76% | 10% | 74% | 18% | | | 80% | 23% |
| 9 | 96% | 6% | | | 95% | 3% | 71% | 25% |
| 10 | 76% | 21% | 59% | 29% | | | 71% | 31% |
| 11 | 76% | 21% | 59% | 29% | | | 66% | 30% |

## 4. Integration of Statistics into Senior Unit Operations Lab

One measure of the effectiveness of the statistics class is the degree to which students then apply these concepts to the real measurements they are taking in senior lab. In fact, the lab problem statements typically contain statements such as,

> Your final, written report should include as a minimum:
>
> 3. ...
> 4. Use of appropriate statistical methodologies for data analysis and interpretation.

While a systematic analysis has not been undertaken, some historical evidence can be presented. It is hoped a more careful analysis from the 2003 lab can be presented at the meeting. The first quarter of the two-quarter lab sequence (ChE 414) is highly structured and focuses on the students completing 3 unit operation experiments. This second quarter of the senior lab course (ChE 415) builds on the work done in UO Lab 1. The focus is on working independently, developing a project proposal, completing experimental work and writing a final technical memorandum that includes recommendations for future work.

During 2001, in serving as a technical consultant for the microelectronics related labs in ChE 415, and discussing the statistical analysis of the students throughout the year with the lab instructor, it was determined that the use of statistical methods needed to be enhanced. Review of the written reports confirmed this belief. Thus, two "refresher" lectures of ChE 302 were included in ChE 414 during W 2002. One of the three experiments asked students to compare if different levels of several factors (velocity, flow orientation and inclusion of a steam trap) had an effect on the performance of a heat exchanger. This study is amenable to ANOVA, and was the most sophisticated statistical analysis needed for the three labs. Therefore, the first refresher lecture, focused largely on a review of ANOVA. The good news is groups correctly used ANOVA in analysis, and many groups demonstrated impressive statistical analysis for the heat exchanger experiment. The extent of statistical analysis rose notably throughout the course. The bad news is that some groups used ANOVA even when it was not needed and in fact, relied on its conclusions even when they contradicted common sense. This symptom seemed more evident in the open-ended assignments in the second term. It seems the pendulum had swung too far! This year there will be added emphasis on using judgment when applying statistical methods.

## 5. Summary

The approach to incorporating statistics in the ChE department at OSU has been discussed. Most of the content is delivered in a one term required class, *Chemical Process Statistics.* The use of industrial case studies reinforces learning. Assessment from the industrial advisory board, MECOP interns and a multi-faceted in class process demonstrate this class is effective. Review of the material in senior lab helps reinforce the use of statistical concepts. However, care must be taken to make sure students use solid engineering judgment in applying statistical methods.

## 6. Acknowledgments

Without the help, support and commitment of industry this course could never have been developed. The author is especially grateful to Michael Matthews who integrated the class into the Merix statistical process control software. Information from, support of, and technical discussions with Adrian Kriz and Lori Thompson of Hewlett-Packard and Mike McMaster, Eric Warninghoff, and Ian Reuf of Merix have also been of great value. The author is grateful for partial support provided by the Intel Faculty Fellowship Program.

### 7. References

1. Shunta, Joseph P., “Use Statistics to Identify Process Control Opportunities,” Chem. Eng Progress p. 47 October 1996.
2. Roger E. Eckert, “Applied Statistics: Are Educators Meeting the Challenge,” Chem. Eng. Ed. p. 122 spring 1996.
3. Wheeler, James M., “Getting Started: Six-Sigma Control of Chemical Operations,” Chem. Eng Progress p. 76 June 2002.
4. Trivedi, Yogesh B., Applying 6 Sigma,” Chem. Eng Progress p. 76 July 2002.
5. Deshpande, Pradeep B., Sohan L. Makker and Mark Goldstein, “Boost Competitiveness via Six Sigma,” Chem. Eng Progress p. 65 September 1999.
4. Anderson, Mark J. and Patrick J. Whitcomb, “Optimize Your Process-Optimization Efforts,” Chem. Eng Progress p. 51 December 1996.
7. Fahidy, Thomas Z., “An Undergraduate Course in Applied Probability and Statistics,” Chem. Eng. Ed. p. 170 spring 2002.
8. Dianne Dorland and K. Karen Yin, “Teaching Statistics to ChE Students,” Chem. Eng. Ed. 170 summer 1997.
9. McConica, C, *Self-Study Report for Chemical Engineering at Oregon State University* (2002).
10. Ayyub, B.M., and R.H. McCuen, *Probability, Statistics, and Reliability for Engineers and Scientists*, Chapman & Hall/CRC, Boca Raton, FL. (2003).
11. Hayter, A.J., *Probability and Statistics for Engineers and Scientists*, 2nd Ed., Duxbury Press, Pacific Grove, CA (2002).
12. Johnson, R.H., *Miller and Freund’s Probability and Statistics for Engineers*, 6th. Ed, Prentice Hall, Upper Saddle River, NJ (2000).
13. L.L. Lapin, *Modern Engineering Statistics*, Duxbury Press; Belmont, CA (1997).
14. Levine, D.M., P.P. Ramsey, and R.K. Smidt, *Applied Statistics For Engineers and Scientists Using Microsoft Excel and MINITAB*, Prentice Hall, Upper Saddle River, NJ (2001).
15. Montgomery, D.C., G.C. Runger, and N.F. Hubele, *Engineering Statistics*, John Wiley & Sons, New York, NY (1998).
16. Vining, G.G., *Statistical Methods for Engineers*, Duxbury Press, Pacific Grove, CA (1998).
17. Walpole, R.E., R.H. Myers, S.L. Myers, and K. Yee, *Probability and Statistics for Engineers and Scientists*, 7th Ed., Prentice Hall, Upper Saddle River, NJ (2002).
18. Scheaffer, R.L. and J.T. McClave, *Probability and Statistics for Engineers and Scientists*, 4th Ed., Duxbury Press, Belmont, CA (2002).

19. Devore, J. and N. Farnum, *Applied Statistics For Engineers and Scientist*, Duxbury Press, Pacific Grove, CA (1999).
20. Czitrom, Veronica and Karen Horrell, "SEMATECH Qualification Plan" in *Statistical Case Studies for Industrial Process Improvement*, Society for Industrial and Applied Mathematics, Philadelphia, PA; American Statistical Association, Alexandria, VA (1997).
21. International 300 mm Initiative, "Metrology Tool Gauge Study Procedure for the International 300 MM Initiative (I300I)" http://www.sematech.org/public/docubase/document/3295axfr.pdf (1997).
22. Koretsky, M.D., "Applied Statistics: Introducing Analysis of Variance through Measurement System Analysis" Chem. Eng. Ed., *submitted*.
23. Koretsky, M.D., "Applied *Chemical Process Statistics* - Bringing Industrial Data to the Classroom," Proc. ASEE, Session 2213 (1998).
24. Joshi, M. and K Sprague "Obtaining and Using Statistical Process Control Limits in the Semiconductor Industry," in *Statistical Case Studies for Industrial Process Improvement*, V. Czitrom and P. D. Spagon Eds, ASA (1997).
25 Barnett, J., V. Czitrom, P.W.M. John and R.V. Leon, "Using Fewer Wafers to Resolve Confounding in Screening Experiments," in *Statistical Case Studies for Industrial Process Improvement*, V. Czitrom and P. D. Spagon Eds, ASA (1997).


**Milo D. Koretsky** is an Associate Professor of Chemical Engineering at OSU. He received his BS and MS degrees from UCSD and Ph D from UC Berkeley, all in chemical engineering. Professor Koretsky's research interests are in thin film materials processing including: plasma etching, chemical vapor deposition, electrochemical processes and chemical process statistics. His book, *Engineering and Chemical Thermodynamics*, is due out in December 2003.

## Appendix. The Language of Statistics Survey

# The Language of Statistics

ChE 399
Chemical Process Statistics
Prof. Milo Koretsky

**1 = I have no idea what whis word means in the context of applied statistics**
**2 = I have heard this word used, but I am not sure what it means**
**3 = I can understand this word in context, but cannot define it**
**4 = I can define this word**

.= After Class

| | Before | | | After | | | |
|---|---|---|---|---|---|---|---|
| | Mean | Median | Standard Deviation | Mean | Median | Standard Deviation | Δmean |
| mean | 3.64 | 4 | 0.62 | 4.00 | 4 | 0.00 | 0.36 |
| median | 3.39 | 4 | 0.74 | 4.00 | 4 | 0.00 | 0.61 |
| mode | 2.89 | 3 | 1.15 | 3.82 | 4 | 0.50 | 0.93 |
| central tendency | 2.04 | 2 | 1.04 | 3.77 | 4 | 0.53 | 1.74 |
| | | | | | | | |
| standard deviation | 3.15 | 3 | 0.82 | 3.95 | 4 | 0.21 | 0.81 |
| range | 3.46 | 4 | 0.84 | 4.00 | 4 | 0.00 | 0.54 |
| dispersion | 2.44 | 3 | 1.01 | 3.50 | 4 | 0.67 | 1.06 |
| | | | | | | | |
| population | 3.32 | 4 | 0.82 | 3.95 | 4 | 0.21 | 0.63 |
| sample | 3.46 | 4 | 0.69 | 3.95 | 4 | 0.21 | 0.49 |
| attribute variable | 1.64 | 1.5 | 0.73 | 3.36 | 4 | 0.85 | 1.72 |
| continuous variable | 1.61 | 1.5 | 0.69 | 3.41 | 4 | 0.80 | 1.80 |
| | | | | | | | |
| box plot | 1.93 | 2 | 1.00 | 3.95 | 4 | 0.21 | 2.03 |
| histogram | 2.43 | 2.5 | 1.07 | 4.00 | 4 | 0.00 | 1.57 |
| distribution | 3.00 | 3 | 0.98 | 3.95 | 4 | 0.21 | 0.95 |
| normal distribution | 2.54 | 3 | 1.04 | 4.00 | 4 | 0.00 | 1.46 |
| binomial distribution | 1.68 | 1 | 0.90 | 3.73 | 4 | 0.55 | 2.05 |
| Poisson distribution | 1.32 | 1 | 0.55 | 2.50 | 2 | 0.96 | 1.18 |
| student t distribution | 1.29 | 1 | 0.60 | 3.50 | 4 | 0.60 | 2.21 |
| | | | | | | | |
| variation | 3.00 | 3 | 0.86 | 4.00 | 4 | 0.00 | 1.00 |
| common causes | 1.71 | 2 | 0.71 | 4.00 | 4 | 0.00 | 2.29 |
| special causes | 1.68 | 2 | 0.72 | 4.00 | 4 | 0.00 | 2.32 |
| | | | | | | | |
| true value | 1.86 | 1.5 | 0.97 | 3.45 | 4 | 0.91 | 1.60 |
| accuracy | 3.14 | 3 | 0.89 | 3.86 | 4 | 0.35 | 0.72 |
| precision | 3.07 | 3 | 0.90 | 3.86 | 4 | 0.35 | 0.79 |
| repeatability | 2.93 | 3 | 0.86 | 3.86 | 4 | 0.47 | 0.94 |
| reproducibility | 2.89 | 3 | 0.83 | 3.86 | 4 | 0.47 | 0.97 |
| stability | 2.64 | 3 | 0.95 | 3.45 | 4 | 0.80 | 0.81 |
| precision to tolerance ratio | 1.57 | 1 | 0.84 | 3.82 | 4 | 0.50 | 2.25 |
| | | | | | | | |
| confidence interval | 1.50 | 1 | 0.75 | 4.00 | 4 | 0.00 | 2.50 |
| hypothesis testing | 2.04 | 2 | 1.07 | 2.95 | 3 | 1.05 | 0.92 |
| null hypothesis | 1.57 | 1 | 0.84 | 2.73 | 3 | 1.03 | 1.16 |
| alternative hypothesis | 1.61 | 1 | 0.92 | 2.55 | 3 | 1.06 | 0.94 |
| | | | | | | | |
| control chart | 1.71 | 1 | 0.94 | 3.86 | 4 | 0.35 | 2.15 |
| control limit | 1.68 | 1 | 0.86 | 3.95 | 4 | 0.21 | 2.28 |
| specification limit | 1.61 | 1 | 0.96 | 3.95 | 4 | 0.21 | 2.35 |
| out of control | 2.14 | 2 | 1.11 | 3.91 | 4 | 0.29 | 1.77 |
| run | 2.19 | 2 | 0.88 | 3.90 | 4 | 0.30 | 1.72 |
| trend | 2.59 | 3 | 1.05 | 3.86 | 4 | 0.36 | 1.26 |
| capability | 1.96 | 2 | 0.76 | 3.67 | 4 | 0.58 | 1.70 |
| $C_P$ | 1.30 | 1 | 0.67 | 3.76 | 4 | 0.44 | 2.47 |
| $C_{PK}$ | 1.11 | 1 | 0.32 | 3.76 | 4 | 0.44 | 2.65 |
| | | | | | | | |
| Analysis of Variance (ANOVA | 1.26 | 1 | 0.59 | 3.81 | 4 | 0.40 | 2.55 |
| F Test | 1.11 | 1 | 0.42 | 3.57 | 4 | 0.60 | 2.46 |
| | | | | | | | |
| Design of Experiment (DOE) | 1.74 | 1 | 0.94 | 3.86 | 4 | 0.36 | 2.12 |
| factor | 1.81 | 1 | 1.04 | 3.71 | 4 | 0.56 | 1.90 |
| level | 1.67 | 1 | 0.92 | 3.67 | 4 | 0.58 | 2.00 |
| full factorial design | 1.22 | 1 | 0.64 | 3.81 | 4 | 0.40 | 2.59 |
| fractional factorial design | 1.22 | 1 | 0.64 | 3.76 | 4 | 0.44 | 2.54 |
| Screening design | 1.19 | 1 | 0.48 | 2.43 | 3 | 1.03 | 1.24 |
| interaction | 1.56 | 1 | 0.85 | 3.71 | 4 | 0.46 | 2.16 |
| **Total** | 104.36 | 102.5 | 25.11 | 182.41 | 186.5 | 17.30 | 78.05 |
| | 2.11 | | | 3.69 | | | 1.58 |

# Integration of Statistics Throughout the Undergraduate Curriculum: Use of the Senior Chemical Engineering Unit Operations Laboratory as an End-of-Program Statistics Assessment Course

**Michael E. Prudich, Darin Ridgway, Valerie L. Young**
**Department of Chemical Engineering**
**Ohio University**

## Abstract

Graduates of chemical engineering programs should have the ability to use basic statistical techniques to analyze and interpret process and experimental data. Chemical engineers should also have the ability to identify the statistical appropriate tool to accomplish a specific data analysis task. This paper describes the undergraduate course of instruction in statistics in the chemical engineering program at Ohio University. This course of instruction begins with a basic introduction to linear regression which takes place in a freshman-level chemical engineering computing course, includes a junior-level course devoted to statistical analysis and experimental design, and ends with the application of statistical data analysis tools in the senior-level unit operations laboratory course sequence. Spreading the exposure to statistical analysis across the curriculum permits longitudinal reinforcement of important concepts and skills. Special emphasis is placed on the use of the senior-level unit operations laboratory experience as a capstone statistics usage and assessment tool. This course provides the students with an opportunity to break out of the "chapter box" which is often characteristic of stand-alone statistical methods courses. The courses also emphasize the point that experimental planning and design includes not only the selection of the experimental parameters to be studied, but also the planning of the data analysis and statistical treatments to be utilized in the interpretation of the experimental data that is acquired. The proper assessment of statistics-related performance in the senior-level laboratory courses provides end-of-program assessment data on student statistical skills and abilities.

## Statistics Throughout the Curriculum

Statistics are used by chemical engineers both to interpret data and to formulate and test models derived from data. The importance of statistics in the engineering curriculum has been recognized by ABET. The 1999-2000 general criteria[1] include the requirement that "students must demonstrate knowledge of the application of probability and statistics to engineering problems." Recently, with the advent of EC 2000, chemical engineering departments have had more flexibility in determining learning objectives and their associated outcomes. The need for a working knowledge of statistical analysis tools and techniques is specifically recognized by the Department of Chemical Engineering at Ohio University under our Objective 1: "Graduates will have a strong foundation in chemical engineering theory and practice" as Outcome 1.d. "Students will demonstrate the ability to formulate models, design and conduct experiments, analyze data, and interpret results using statistical tools."

The overall exposure to statistics in the Department of Chemical Engineering at Ohio University has evolved in an ad hoc manner. (Typical use and/or instruction of statistics and statistics-related subjects in required courses in the chemical engineering curriculum is described in Table 1.) There was no undergraduate course dedicated to statistics required for graduation in the chemical engineering curriculum prior to 1996. Before that time, statistics instruction consisted of what individual instructors would decide to pass along in their courses depending on their particular desires and course needs, as well as a brief introduction to linear and nonlinear regression and, sometimes, hypothesis testing, that was included in the last 10-20% of a junior-level course in chemical engineering calculations/computer solutions. Beginning in 1992, an undergraduate chemical engineering technical elective course in experimental design and statistics was offered. This undergraduate course in experimental design and statistics was elevated from technical elective to required status in the chemical engineering curriculum beginning in 1996.

Our required undergraduate course in experimental design and statistics for chemical engineers resembles courses typically offered by other universities. It is a stand alone, lecture-type course that is not explicitly related to any of the other courses in our curriculum. A standard text[2], was used for the required statistics course in 2001. The subjects covered in this course are the usual ones including: probability; random variables, the normal distribution; sample distributions; determining an adequate sample size; various types of hypothesis testing; fitting equations to data; factorial analysis; and an introduction to statistical process control. Supplementary materials are provided which cover nonlinear least squares analysis and fractional factorial experimental design. Most would agree that the subjects listed above are important for chemical engineers and most students perform well enough within the limits of the course (specifically, Spring Quarter junior year). Later assessments (see below) indicate that there are three deficiencies in presenting this material in a 'stand alone' matter: (1) the students learn the material in the context of individual

Table 1. Statistics Usage and/or Instruction in the Undergraduate Chemical Engineering Curriculum at Ohio University.

| Course | Use of and/or instruction in statistics. |
|---|---|
| Freshman Year (Year #1) | |
| ChE 100 - Introduction to Chemical Engineering | None reported. |
| ChE 101 - Approaches to Chemical Engineering Problem Solving | How to linearize a non-linear algebraic equation. The use of MATLAB to determine equation parameters using single and multiple linear regression. Residuals are calculated. |

| Table 1. Statistics Usage and/or Instruction in the Undergraduate Chemical Engineering Curriculum at Ohio University (continued). | |
|---|---|
| Course | Use of and/or instruction in statistics. |
| **Sophomore Year (Year #2)** | |
| ChE 200 - Material Balances | None reported. |
| ChE 201 - Energy Balances | Introduction to the concept that uncertainty in measurements to show that real energy balances do not close to the nth decimal place. The students record temperatures and flow rates through a simple concentric tube heat exchanger, calculate the heat loss from the hot side and the heat gain by the cold side, compare the numbers, and observe how the comparison changes if the value of each temperature changes 1 °C. No formal calculations of uncertainty are made. |
| **Junior Year (Year #3)** | |
| ChE 305/306 - Thermo; ChE 345 - Fluids; ChE 347 - Mass Transfer and Separations; ChE 400 - Applied ChE Calculations | None reported. |
| ChE 307/308 - Chemical Engineering Kinetics | Least squares fitting is used. Linearize rate laws and determine rate parameters through least squares fitting. The students are introduced to non-linear least-squares analysis through a section in the course text and a homework assignment. |
| ChE 346 - Heat Transfer | An associated laboratory experiment. Experimental design - the concept of spreading out the values of independent variables. Estimating the error in a calculation. Linearizing non-linear algebraic equations. Determination of equation parameters by least-squares analysis. Parity plots. |
| ChE 408 - Engineering Experimental Design | Statistics course (details in text). |
| **Senior Year (Year #4)** | |
| ChE 443/444 - Process Design; ChE 448 - Safety in the Process Industry; ChE 499 - Chemical Engineering Senior Assessment | None reported. |
| ChE 415/416 - Unit Ops. Lab | Details below. |
| ChE 417 - Process Controls Laboratory | linear and/or non-linear least-squares analysis are used for three lab projects. Hypothesis testing is used for at least three lab projects. |
| ChE 442 - Process Controls | least squares fitting is used to determine the time constant and/or steady-state gain from response data. |

‘chapter boxes’ (for example, the problems at the end of Chapter 4 will be on the subject matter covered in Chapter 4), (2) the students have no professional motivation or practical understanding of the chemical engineering uses of the subject of the subject matter being presented (most examples are not chemical engineering related), and (3) too much time elapses between formal instruction in the use of statistics for experimental planning and data analysis and the practical need to use statistics for experimental planning and data analysis.

**Assessment Via the Senior-Level Unit Operations Course**

The unit operations laboratory at Ohio University is a two-quarter sequence taken during the senior year. Four experiments are performed each quarter. The junior-year thermodynamics, reactor design, and unit operations courses are prerequisites, along with a course in experimental design and statistics. Students typically work in teams of three. Each experiment is mentored by a different faculty member. The faculty mentor is responsible for all aspects of assigning of objectives, meeting with the students and grading the reports. The format includes an informal meeting with the faculty mentor, a graded prelab in which the students are to present their experimental plan and expected data analysis plan, one week (10 hours) of lab time and a final written report. Grading is on common course grading rubrics for both the prelab and postlab reports. These grading rubrics are supplied to the students during an orientation held at the first class meeting and are available throughout the quarter on the course webpage[3]. The details of these grading sheets and the philosophies behind their use have been described elsewhere[4]. The grading sheets cover all aspects of the prelab and postlab reports including both format and content with the portions related to experimental design and statistical analysis being those of interest to this study.

The unit operations laboratory sequence as offered at Ohio University can be thought of as having two functions related to experimental design and statistics: (1) the reinforcement of analysis skills learned by students in previous courses, and (2) the evaluation and assessment of that previous learning. The two quarter unit operations laboratory sequence is ideally situated and constituted for the assessment of the students’ abilities to apply their training in statistics due to both its location in the senior year of the curriculum and to the course requirement to identify the need for and correctly apply a wide range of statistical techniques to the completion of the data analysis that comes about as the result of the experimentation (see Table 2). Every quarter in this course each lab team prepares four prelab and four postlab reports, and each individual prepares one design memorandum and makes one oral presentation describing their design. Each of these assignments is evaluated with respect to a number of Primary Traits (10 each for the Prelab and Postlab reports and five for the Design Memorandum and six for the Design Oral) many of which include elements related to the application of statistics and/or experimental design (see Table 3). For the prelab and postlab reports, a substantive deficiency in any Primary Trait triggers a mandatory re-write of the report.

| Table 2. Experiments Performed in Unit Operations Laboratory With Associated Potential for Statistical Analysis (Fall Quarter AY2002/2003). | |
|---|---|
| Experiment | Potential Applications of Statistics |
| Drying | • experimental design<br>• linear regression (to find slope of line; find the confidence interval around the slope; to find the intercept; find the confidence interval around the intercept; to determine linearity of data; to find the equation parameters for the linearized version of a nonlinear heat-transfer and mass-transfer coefficient relationships; to find the confidence intervals around the equation parameters)<br>• nonlinear regression (to find equation parameters for nonlinear heat-transfer and mass-transfer coefficient relationships; to find the confidence intervals around the equation parameters)<br>• hypothesis testing (paired t-test to compare if two measurement techniques yield the same values; t-test to determine if experimental measurements agree with literature/theoretical values; t-test to determine if equation factors are significant) |
| Fluid Flow | • hypothesis testing (t-test to determine if experimental measurements agree with literature/theoretical values) |
| Membrane Permeation | • experimental design<br>• linear regression (to find slope of line to determine the Fick's Law parameter; find the confidence interval around the slope; to find the intercept; find the confidence interval around the intercept; to determine linearity of data)<br>• hypothesis testing (paired t-test to determine if countercurrent and cocurrent configurations produce the same results; t-tests on slopes to determine if the Fick's Law parameter is a constant.) |
| Mixing | • experimental design<br>• linear regression (to find the equation parameters for the linearized version of nonlinear mixing power and performance relationships; to find the confidence intervals around the equation parameters)<br>• nonlinear regression (to find the equation parameters for nonlinear mixing power and performance relationships; to find the confidence intervals around the equation parameters)<br>• hypothesis testing (t-test to determine if experimental measurements agree with literature/theoretical values; t-test to determine if equation factors are significant) |

| Table 3. Primary Traits Which Include Some Aspect of Statistics. | |
|---|---|
| Primary Trait Name | Description of Statistics Content |
| Prelab Report (2 of 10 Primary Traits include statistical content) | |
| *Experimental Test Matrix* | "Specifies all experimental conditions with reasonable values, sufficient range to meet all objectives, reasonable number of runs, repeat runs..." |
| *Statistical Methods* | "Uncertainties for all values stated. Uncertainties appropriately based on propagation of experimental error or on statistical methods. States how error will be calculated. Planned evaluation of model equations based on variance, correlation coefficient, and/or residual plots. Comparison of values based on hypothesis testing (e.g. t-test) as appropriate. Specific." |
| Postlab Report (4 of 10 Primary Traits include statistical content) | |
| *Report Overall, Experimental Uncertainty* | "All values quoted with uncertainties throughout the report. Uncertainties appropriately based on propagation of experimental error or on statistical methods. Includes error bars on figures as appropriate." |
| *Presentation Style* | "... Error bars are included where appropriate." |
| *Statistical Analysis* | "Complete and appropriate statistical analysis included. Evaluation of data, results, and/or model equations based on variance, correlation coefficient, and residual plots. Comparison of values based on hypothesis testing (e.g. t-test) as appropriate. Appropriate error bars on all figures." |
| *Conclusions* | "... Expresses limitations in terms of uncertainties ..." |
| Design Oral Critique (2 of 6 Primary Traits include statistical content) | |
| *Experimental Approach* | "... I (the evaluator) know the uncertainty on each of the experimentally-determined parameters." |
| *Proposed Design* | "... specifications which depend on experimentally-determined values are given an appropriate range of values based on experimental uncertainty..." |
| Design Memo Grading (2 of 5 Primary Traits include statistical content) | |
| *Description of Experimental Approach* | "... uncertainty stated on each experimentally-determined parameter..." |
| *Proposed Design* | " ... specifications which depend on experimentally-determined values are given an appropriate range of values based on experimental uncertainty..." |

Prelab report grading results for Unit Operations Laboratory I were consistent for Academic Years 2001/2002 and 2002/2003 with problem areas identified with the two Primary Traits *(Data Analysis/Statistical Methods* and *Data Analysis/Expected Data and Results*) associated with the analysis of experimental data after it has been collected. During Fall Quarter AY2001/2002, ten lab teams participated in ChE 415 - Unit Operations Laboratory I. Of a total of 40 prelab reports submitted, 22 (55%)were returned for mandatory re-writes due to substantive deficiencies in at least one of the Primary Traits. Of the 22 mandatory re-writes, nine were returned for deficiencies found in the *Data Analysis/Statistical Methods* category. This number represents 41% of the mandatory prelab rewrites and 23% of the total prelab reports. The next highest category that resulted in mandatory re-writes was the Primary Trait *Data Analysis/Expected Data and Results* which resulted in five mandatory rewrites (23% of all mandatory prelab rewrites; 13% of all prelab reports submitted).

Nine laboratory teams were enrolled in Unit Operations Laboratory I Fall Quarter AY2002/2003. Of the 36 prelab reports submitted during this quarter, only nine (25%) were returned for mandatory re-writes. Of these nine mandatory re-writes, six (67% of the mandatory re-writes; 17% of all prelab reports) were returned for deficiencies found in the *Data Analysis/Statistical Methods* category. As in AY2001/2003, the next highest category that resulted in mandatory re-writes was the Primary Trait *Data Analysis/Expected Data and Results* (44% of the mandatory re-writes; 11% of all prelab reports). [Note: Percentages may sum to greater than 100% since an individual report may fail more than one Primary Trait.]

Prelab reports are especially important in that they provide the roadmap by which the students acquire and analyze their data. A poor prelab report often results in deficiencies in data acquisition (wrong kind and/or not enough data taken) and in data analysis (inappropriate methods and/or incomplete analyses).

Difficulties identified in the prelab reports included:

* no statistical analyses planned
* vagueness (for example, “Appropriate statistical analyses will be performed.”)
* statistical analyses not planned for all appropriate situations (at least one situation properly addressed with, perhaps, the assumption made that only one application of statistics will successfully get that ‘box’ checked on the grading sheet)
* inappropriate statistical methods proposed (To a student with a new hammer, every problem is a nail - “I know how to do the t-test, I will apply it to every problem.”)
* insufficient planning for data collection (even if the correct statistic methods are proposed, the students often make their experimental plan without considering the type, range, and number of experimental trails required in order to provide the data necessary to their chosen methods of analysis)

Upon completion of each experiment, a laboratory team would prepare and submit a postlab report documenting their experimentation, providing an analysis of their data, and describing any models or correlations made for the purpose of later design work. Of the total 40 postlab reports submitted for Fall Quarter AY2001/2003, only six (15% of the total number of postlab reports submitted) reports included deficiencies severe enough to result in mandatory rewrites. Only one of the six mandatory rewrites was for a deficiency related to statistical analysis (17% of all postlab rewrites; 3% of all postlab reports).

Once again, data collected for AY2002/2003 show trends similar to those observed for AY2001/2002. For the 36 postlab reports submitted, only four (29% of all postlab re-writes; 11% of all postlab reports) required re-writes for a deficiency related to statistical analysis, a decrease from the percentage observed for prelab reports submitted by the same cohort of students.

Our tentative conclusion is that students are usually able to perform the mathematical manipulations associated with the required statistical analysis tools as evidenced by their success with the postlab reports. However, our students were often unable, or unwilling, or fail to see the necessity to plan the data analyses they will need and to select the statistical tools required to make those analyses. This problem was evidenced by the number of statistics-based mandatory prelab rewrites. The postlab reports were successful because the data analysis tools had already been identified as a result of the required prelab rewrites and subsequent consultation with the faculty mentors. Another way of describing this problem would be that the students had learned how to use the individual tools in their tool boxes in the context of the "chapter boxes" in which they were presented in the required statistics course. When confronted with an undefined problem, the students would have their tool boxes, but lack the confidence and experience to select the correct statistical tool for the job required.

The students enrolled in Unit Operations Laboratory I perceived a deficiency in their prior instruction in statistical methods as evidenced by end-of-course survey results. As a part of the end-of-course survey the students are asked to evaluate prerequisite chemical engineering courses as to their adequacy ("5" being prepared "very well", "1" being prepared "poorly") in preparing them for the unit operations laboratory experience. Of the eight courses/subject areas evaluated, only one (experimental design and statistics) has consistently received an unfavorable rating, 2.0 (next lowest was 4.4) for AY2001/2002 and again 2.0 (next lowest was 4.3) for AY2002/2003. It is our hypothesis that the problem is not in the subject matter covered within the course that we offer, rather it is the method of coverage and the lack of timely, significant reinforcement of statistical data analysis skills.

**Recommended Solutions**

The identification of deficiencies without plans to address those deficiencies is of little use. We are implementing the following changes in our overall plan of instruction for experimental design and statistics.

*The required experimental design and statistics course will be modified to be taught more from a case study perspective.* (This requires the instructor to formulate and/or acquire relevant cases. The course[5] has recently been modified and offered in this manner[5] Winter Quarter AY2002/2003. A detailed description of the course can be found at the course website[6].)

*The required experimental design and statistics course will include more 'chemical engineering' examples and problems.* (This requires the instructor to formulate and/or acquire relevant problems. The course[5] has recently been modified and offered in this manner Winter Quarter AY2002/2003.)

*Timely reinforcement (in terms of class usage and experimentation and data analysis) will be provided for the materials covered in the required experimental design and statistics course.*

The chemical engineering faculty at Ohio University have already experienced success with embedding computer programming (MATLAB) throughout the curriculum. Prior to this embedding our students learned MATLAB their freshman year, but were not required to use it again until their junior year in the curriculum. It was unsurprising that they entered the junior-year class with little or no retention of their MATLAB skills. This problem was solved by embedding problems requiring the use of MATLAB in additional sophomore- and junior-year courses. Evidence of the success of this approach is the fact that many students will solve simple and complex problems using MATLAB programming during senior year without be required to do so. They have taken ownership of this skill. A similar approach should be taken in the area of statistical skills.

An initial approach will be to add opportunities for students to apply the skills learned in the required undergraduate statistics course in other core chemical engineering courses. A paper by Nelson and Walloons[6] have endorsed this approach stating that "we believe that an introductory course in statistics for engineers must be considered in conjunction with the entire engineering curriculum. No matter how good an introductory course in statistics is, if students are not asked to use this material in any subsequent courses, they will soon forget it and most probably question why they were required to take the course in the first place. Thus, we propose to enlarge our area of concern from just an introductory course in statistics to how the concepts from this course can be utilized, reinforced, and enhanced in subsequent engineering courses." Their initial application of this concept was to undergraduate chemical engineering unit operations laboratories at Clemson University. They further state that, "Our approach will be to start by introducing statistical techniques into the engineering labs. We are suggesting engineering labs as the initial point of attack because laboratories are the setting where students perform experiments and collect data." Laboratory exercises were selected for reinforcement because: "This approach combines the advantages of hands-on activities and real world data sets. Students create their own data sets, which gives them added motivation to do the analyses. Moreover, because the data come from an experiment whose main purpose is to illustrate important engineering principles, the data carry a sense of real import that statistics courses ordinarily can achieve only with archival data[7]."

Two courses have been identified for use in reinforcing experimental design and statistics learning: ChE 346 - Heat Transfer and ChE 347 - Mass Transfer and Separations. The heat transfer course is offered the same quarter (Winter Quarter) as the required undergraduate statistics course and already has a laboratory project incorporated into the course. An effort has been made to integrate this existing laboratory project with the subject matter being presented in the statistics course. The first offering of the integrated heat-transfer experiment was made in Winter Quarter AY2002/2003. The second course, the Mass Transfer and Separations course, is offered the quarter directly following the statistics course (Spring Quarter) and does not currently have a laboratory project. We plan to move the membrane permeation experiment (see Table 2, above) from the senior Unit Operations Laboratory course and, with some slight modifications, introduce it as a laboratory project in the mass transfer and separations course. The first offering of the integrated mass-transfer experiment will be made in Spring Quarter AY2002/2003. Cobb[8] has recognized the advantages of this practicum-type approach of instruction. Cobb lists the following perceived advantages of hands-on activities in instruction in statistics (1) "the resulting data are fresh, not someone else's leftovers, (2) students are actively involved in data production" - (a) "the student who has a hand in creating a data set is nearly always motivated to analyze it, (b) experience with variability is immediate and concrete," and (c) "the activities can involve students with design (of experiments) issues."

We are hopeful that, after making the changes described above, we will be able to assess and report improved learning in the areas of experimental design and statistics for chemical engineering students at Ohio University.

Bibliographic Information


1. ABET, 1998, "Criteria for Accrediting Engineering Programs, Effective for Evaluations During the 1999-2000 Accreditation Cycle."
2. Barnes, J.W., 1994, Statistical Analysis for Engineers and Scientists: A Computer-Based Approach, McGraw-Hill, New York.
3. Course materials for our two unit operations laboratory courses, including copies of the grading rubrics, can be reached through at http://webche.ent.ohiou.edu/course_index.html
4. Young, V.L., Ridgway D., Prudich, M.E., Goetz, D.J., and Stuart, B.J., 2001, "Criterion-Based Grading for Learning and Assessment in Unit Operations Laboratory," Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition, Session 3213.
5. Young, V.L., 2003, "Designing a Statistics Course for Chemical Engineers," Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition, Session 3513.
6. The homepage for our undergraduate chemical engineering statistics course can be found at http://webche.ent.ohiou.edu/che408/homepage.html
7. Nelson, P. and Walloons, T., 1993, "Improving the Undergraduate Statistical Education of Engineers," Journal of Statistics Education, v. 1, n. 1, http://www.amstat.org/publications/jse/v1n1/cobb.supp.nelson.html
8. Cobb, G.W., 1993, "Reconsidering Statistics Education: A National Science Foundation Conference," Journal of Statistics Education, v. 1, n. 1, http://www.amstat.org/publications/jse/v1n1/cobb.html

Biographical Information

MICHAEL E. PRUDICH is Professor and Chair of the Department of Chemical Engineering at Ohio University. His B.S., M.S., and Ph.D. degrees are from West Virginia University. He has been a member of the faculty at Ohio University for 18 years, joining academia after working for Gulf Oil Company in Pittsburgh, PA.

DARIN RIDGWAY is an Associate Professor of Chemical Engineering at Ohio University. He received his B.S. from Rose-Hulman Institute of Technology, M.S. from Louisiana Tech University, and Ph.D. from Florida State University. His current research interests are in the areas of biochemical engineering.

VALERIE L. YOUNG is an Assistant Professor of Chemical Engineering at Ohio University. She received her B.S. from Lehigh University and Ph.D. from Virginia Tech University. Her current research interests are in the areas of atmospheric chemistry.

**The Use of Active Learning in Design of Engineering Experiments**

Gerardine G. Botte
Ohio University
183 Stocker Center
Athens, OH 45701

This paper discusses the issues and experiences of developing an active learning atmosphere during a Design of Engineering Experiments course. The course covered three main topics: introduction to statistics, design of experiments and statistical process control. Twelve undergraduate students at the sophomore and junior levels participated in the course. The course was taught at the University of Minnesota Duluth. A highly motivated classroom environment was achieved by using a combination of the following techniques: real life examples, classroom projects (individual and group), brainstorming, computer guided sessions, and a special interest course project. The special interest project used hobbies of the students to enlarge their enthusiasm for the course, for instance, one of the students worked in a project to use fractional factorial design to improve her performance in her hammer throw competition; another student used the same technique to improve her performance when playing tennis. Examples of the case studies developed for the course, classroom and take home projects will be presented and discussed, including their impact on the students. Some of the special interest projects developed by the students will be shown and discussed.

## Introduction

The idea of creating an enthusiastic learning atmosphere in the classroom is the dream of any teacher. Of course, that is a dream that depends upon many factors: the enthusiasm of the professor, the motivation of the students, the number of students in the class, and the difficulty of the content covered in the course. Nevertheless, there are some general strategies and tips that can be used to create a keen atmosphere for learning in the classroom. These strategies form part of what is called “active learning”. Traditionally, it is expected that students will be involved in active learning by listening in the lectures and doing some projects out of the classroom that will make them use the concepts learned in class. This conventional way of learning is driven by the constraint in the time of lecture period and by the fact that the student should demonstrate his/her interests for learning.

However, the research literature suggests[1] “that students must do more than just listen. They must read, write, discuss, or be engaged in solving problems. Most important to be actively involved, students must engage in such higher-order thinking tasks as analysis, synthesis, and evaluation. Within this context, it is proposed that strategies promoting active learning be defined as instructional activities involving students in doing things and thinking about what they are doing.”

Basically, it is suggested that the lecture time is divided so that the students can do all these activities (read, write, discuss, and be engage in solving problems) in the classroom. The instructor serves as a mentor and the students learn by doing small projects in the classroom. Additionally, classroom activities must be complemented with longer assignments out of the classroom.

Research[2] has shown that the involvement of students in the educational process through active learning makes students recognize and accept their responsibility for lifelong learning and continued education, which is consistent with ABET 2000 accreditation criteria.[3]

Seeler *et al.*[2] have suggested ways to modify the lecture in order to achieve an active learning atmosphere in the School of Veterinary which have demonstrated excellent results for their students. The objective of this paper is to discuss some of the application of their suggestions along with other tips to achieve active learning in a Design of Engineering Experiments Class for Chemical Engineers. Some examples of student cases studies will be presented.

**Course Description and Scenario**

The Design of Engineering Experiments course was taught by Dr. Botte at the University of Minnesota Duluth in spring 2001. It was Dr. Botte's first course taught. The description of the course according to the UMD course catalog is[4] "CHE 2011. Design of Engineering Experiments: Basic theories of experimental design, data analysis, and statistical process control, emphasizing their application to chemical engineering practice." The freshman introduction to calculus courses (limits, derivatives, integrals, vectors, partial derivatives, etc) are prerequisites of the course. Twelve undergraduate students at the sophomore and junior levels participated in the course. The course was taught twice a week with a lecture time of 1:15 minutes each session, during fifteen weeks (semester format).

**Structure and Course Content**

The first step to implement active learning requires a critical evaluation of the course, its structure and content. The structure and content should be consistent with the educational objectives of the institution, the relationships of the course with others in the curriculum, and the instructor's expectations of the course.[2] The instructor must recognize what is important that students learn from the course (instructor's expectations), which must be related to the application of the course material to the future job of the student. For doing this, is extremely important to use the industrial experience of the instructor and/or to discuss the ideas with the Industrial Advisory Board of the Department. Keeping all this information in mind the objective of the course was to teach the most basic principles and techniques of experimentation, data analysis and statistical process control with a minimum of statistical formality abstraction. Special emphasis was placed on experimentation for quality improvement. Table I presents a summary of the major content covered in the course.

**Table I. Major Content Covered in the Design of Engineering Experiments Course**

| Topic | Details |
|---|---|
| Basic Statistics | 1. Description of variation (e.g. histograms, standard deviation, etc)<br>2. Probability distributions (e.g., Poisson, normal probability distribution, etc) |
| Design of Engineering Experiments (DOE) | 1. Meaning of Quality and quality philosophies<br>2. Full factorial design (two level experiments)<br>3. Fractional factorial design (two level experiments)<br>4. Evaluating variability<br>5. Blocks effects<br>6. Process Optimization with DOE |
| Statistical Process Control | 1. Methods and Philosophies of control charts<br>2. Control Charts for variables<br>3. Control Charts for attributes<br>4. Process Capability |

**Active Learning Implementation**

Concepts were taught using a combination of the following techniques: short case studies (classroom and/or take home projects), real life examples, brainstorming, computed guided sessions, and a special interest course project. The introduction of new concepts was made through a short lecture follow by a practice. The use of all of these techniques is described next.

Lectures:

The instructor elected to use the lecture as the primary educational technique. The total lecture time (1 hour and 15 minutes) was divided in the following sections: 1. warm up period (2 minutes), 2. review (9 minutes), 3. body of the lecture (30 minutes), 4. classroom practice (25 minutes), and 5. summary (9 minutes).

The *warm up period* was used to break the ice in the classroom and to make the students interact with the instructor. A quick conversation in general topics such as: TV programs, new movies, favorite sports, etc were used as examples. Most of the students participated in the conversation and gained confidence with their classmates and the instructor.

The *review time* was used to summarize the key aspects of the previous lecture. Students were asked to help in summarizing the points. They were allowed to quickly review their notes and bring up key points to the classroom. A time for questions about the covered material was permitted.

The *body of the lecture* was used to introduce new concepts. All the lectures were taught using a power point presentation format, which allowed saving time in the introduction of the concepts. A copy of the presentation was provided to the students as handouts at the beginning of the class. The students were not allowed to take notes while the instructor was speaking and explaining the new concepts. Two to three major concepts were introduced in each lecture session. At the end of each concept, the students were given a time to ask questions and a classroom practice related to the topic was assigned.

*Classroom practices* consisted in exercises designed to apply the new concepts introduced during the class. The classroom practices were made in teams, the class was divided into two teams with six members each. Each team was always the same during the whole semester. The two teams kept competing to finish the exercise first, even though it was not originally planned that way. However, this inherent competition between the teams was favorable for the motivation of the students. The classroom practices substituted the examples explained and solved completely by the instructor. That is, after introducing a new concept the instructor did not solve a problem, instead the students were challenged to use the concepts to try to solve an exercise by themselves and the instructor served as a tutor. At the end of the period the solutions of the two teams were discussed and the whole problem was solved in detailed by the instructor. Once again, the students were not allowed to take notes during this time. Copies of the complete solution of the problem were given to students.

The *summary* of the lecture was used to emphasize the most important aspects of the lecture. The students were asked to help providing the key ideas discussed during the lecture. Time for questions was allowed.

Short Case Studies:

Short case studies were used to exercise the concepts explained in class. Some of them were part of the classroom practices explained above which were done by groups and during the class period. Additionally, two take home case studies were assigned during the semester: 1. the chewing gum exercise, and 2. the helicopter experiment. A description of the cases is given in table II. Both take home experiments were done by teams (same members as class teams). A short report from the group as well as a presentation was required in both cases. The speaker for the presentation was selected randomly to assure the participation of each team members in the project. Initially the students complained about this policy but later they realized that it made a difference in the participation of the students in the assignment.

Real life examples and brainstorming:

It really makes a difference in the attitude and interest of the students when the professor uses phrases such as "this is a real industrial problem …I was involved few years ago…" The students get really interesting and willing to learn and listen about the application of the concepts in the industry. For example, when after explaining the use of cause-effect diagrams in the class, Dr. Botte presented briefly the production process of polyvinyl chloride (PVC), and asked the students to work in class in teams (as described in classroom practices) to build a cause-effect diagram for the formation of fish eyes in the resin. The students needed to brainstorm, think, and analyze to propose causes for the problem. Even though it was the first time the students heard about the PVC process some of the causes discussed by them have been analyzed in the PVC industry.

Computed guided sessions:

Computed guides sessions were used to teach how to use design of experiments software and excel to practice some exercises and case studies with computer requirements. A detailed handout with the exercise was provided, so that the students followed it step-by-step (encouraging self learning). The instructor acted as a tutor. Once they finished reproducing and learning the method through the handout, they were asked to solve an exercise. Their solutions were discussed at the end of the class period and the complete solution was presented by the instructor. The complete solution of the problem was provided to the students at the end of the class as a handout.

**Table II. Take Home Short Case Team Exercises**

<table>
<tr><th>Short Case Team Exercise</th><th>Description</th></tr>
<tr><td>Chewing gum exercise</td><td>Design an experiment to evaluate the influence of the following factors: flavor, meal, and gender on the flavor lasting time of gums. Replicate your experiments once and randomize the trials using the randomizing tables. The factors and the levels are summarized below:
<table>
<tr><th>Factor</th><th>Low Level</th><th>High Level</th></tr>
<tr><td>Flavor</td><td>Fruit Juice</td><td>Double Mint</td></tr>
<tr><td>Gender</td><td>Female</td><td>Male</td></tr>
<tr><td>Meal</td><td>Before</td><td>After</td></tr>
</table>
The response is the flavor lasting time in minutes. Build a response table, plot effects, two-way interaction tables and plots. What effects are real? Determine the settings that maximize the response and estimate the maximum value of the response. Prepare a team report with your results and a 5 minutes presentation. The speaker will be chosen randomly in class; therefore all the members of the team should be prepared for presenting and answering questions.</td></tr>
<tr><td>Helicopter experiment</td><td>Product development at “Duluth’s Toys” is developing a cheap distraction toy for use at restaurants to keep children entertained while waiting for service (also while parents are eating). The toy (a paper helicopter) needs to be simple in design since the children will actually be assembling it using scissors, paper clips, etc. A prototype (basic design) has been developed which satisfies the assembly requirements. A study done using the prototype has discovered that the satisfaction of the costumers (children) is directly proportional to the flight time (in seconds). The basic design for the prototype is given.<br>An engineer of the team (chemical engineer from UMD who took ChE-2011) has suggested optimizing the prototype by studying the following factors:
<table>
<tr><th>Factors</th><th>Low Level</th><th>High Level</th></tr>
<tr><td>Paper</td><td>0.04 lbs</td><td>0.26 lbs</td></tr>
<tr><td>Body Fold Width</td><td>1.5”</td><td>2”</td></tr>
<tr><td>Body Design</td><td>No tube</td><td>tube</td></tr>
<tr><td>Wing Width</td><td>1.5”</td><td>2”</td></tr>
<tr><td>Wing Length</td><td>4.75”</td><td>5.75”</td></tr>
<tr><td>Paper Clip</td><td>No</td><td>Yes</td></tr>
<tr><td>Wing Offset</td><td>No</td><td>Yes</td></tr>
</table>
The engineer suggested performing a preliminary study by using 16 runs. He also suggested that is very important to replicate the data 2 times to reduce the variability of the experiment. Also the experiments should be performed in random order. What factors affect the response (flight time)? Optimize your design. What would you suggest to improve the design? Prepare a team report with your results and a 10 minutes presentation. The speaker will be chosen randomly in class; therefore all the members of the team should be prepared for presenting and answering questions.</td></tr>
</table>

.

<u>Special Interest Project</u>:

The special interest project was used as the final project of the course. The students were asked to choose a topic of their interest as their final design of experiments project. The objective of the project was to propose, design, carry out, analyze, write a report, and prepare a presentation for an experiment of their choice. The only constraint of the experiment was that it had to have at least 16 total runs**.** For example, they could run a replicated $2^3$, or a single of $2^4$, or a fractional factorial in 16 runs, or a fold over design. Examples of the students' projects are given in Table III.

The projects were presented the last week of class. The results were excellent; the enthusiasm of the students for their projects was really high, which also demonstrated their motivation for the class. Most of the students brought samples of their experiments to the class, e.g., videos of the experiment, equipment used, even food sample in some cases.

**Table III. Examples of Special Interests Projects Developed by the Students of the Class**

| Project title | Objective | Factors |
|---|---|---|
| The bouncy ball experiment | Find the conditions for the maximum flight time of the average rubber bouncy ball | Temperature, landing surface, and ball size |
| Paper airplanes in flight | Maximize the distance a paper airplane can fly | Weight of the paper, style of the airplane, height at which the airplane was launch, the force that the airplane was launched with |
| The best throw: optimization of throwing technique | Maximize the distance that the hammer travels in the air | Number of spins, number of warm-ups, handle shape |
| The clay's stress strain test | Find the maximum pressure for the clay to start deforming | Different amounts of cornstarch, water, and baking soda |
| Is all about racket size? | Maximize the distance a tennis ball will travel after hitting a tennis racket | Ball age, number of strings in racket, shock absorber |
| Figure skating: the cutting edge | Maximize the time the skater is in the air during a jump | Jump type, number of revolutions, skater |
| What's popping | Minimize old maids | Pop corn type, heat settings, pan type |

## Conclusion

The examples discussed here to incorporate active learning into a Design of Engineering Experiments course gave excellent results. Most of the time the students got to learn by first time hands-on experience, which increase their motivation for the class. The instructor acted as a mentor. The students got to read, write, discuss, and be engaged in solving problems both in the classroom and out of it.

## References


1. C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom*, George Washington University, Washington DC, 1991.

2. D. C. Seeler, G. H. Turnwald, and K. S. Bull, "From Teaching to Learning: Part III. Lectures and Approaches to Active Learning," *J. of Veterinary Medical Education* **21** (1994).
3. ABET-2002, *"2003-2004 Criteria for Accrediting Engineering Programs-Program outcomes and assessment"*, p.1 www.abet.org
4. UMD, *Chemical Engineering Course Catalog*, www.semesters.umn.edu/dulcat/template/courses.cfm

**Biographical Information**

Gerardine G. Botte is an Assistant Professor of Chemical Engineering Department ad Ohio University and Assistant Director of the Institute for Corrosion and Multiphase Technology. She received her B.S. from Universidad de Carabobo (Venezuela), and her M.E. and Ph.D. from University of South Carolina. She worked for three years as a Process Engineering in a Petrochemical Complex (PEQUIVEN, filial of PDVSA. Venezuela) before going to graduate school. Dr. Botte applies chemical engineering principles for the analysis of electrochemical systems. Her current research interests are in fuel cells, lithium ion batteries, and corrosion.

# VBA – An Exercise for Practicing Programming in the ChE Curriculum

by

**Edward M. Rosen**
**EMR Technology Group**

## Introduction

A programming exercise is described by Shacham et al (1) which is carried out in Matlab (2). The exercise involves the analytical solution of the Redlich-Kwong equation for the compressibility factor and the consequent calculation of molar volume, fugacity coefficient, isothermal enthalpy and entropy departure. The authors indicate that other programming languages (such as C or C++) can also be used. Since Matlab is widely used in academia and is often taught as ***the*** programming language, such a choice is reasonable.

In industry, however, languages such as Matlab may not always be available or accessible. More likely Excel would be available and so teaching and carrying out such an exercise in the macro language of Excel, Visual Basic for Applications (VBA), is a viable alternative (3). The undergraduate chemical engineering student is generally familiar with Excel and anecdotal evidence indicates the student gravitates toward its use after exposure to other languages/systems(4).

## The VBA Program

The VBA program looks very similar to the Matlab program. However, the main program (the spreadsheet) is particularly simple in Excel.

There are some small differences, however between the Matlab and VBA languages in this case.  In VBA the sign function is sgn, sqr is the square root and there is no max or acos function in VBA but there is in the Excel spreadsheet. In VBA to find the maximum of a series of numbers, application.max is used and application.acos is used to find the arccosine. In VBA there is also no case differences so bb is used for b and rr is used for r to be different from B and R(see Figure 1 in Reference (1)).

As in Matlab, a function procedure, RKfunc (1), is written which is called directly from the spreadsheet. Depending on what property is desired, a Kode argument is included in the argument list of RKfunc (Figure 1).

Kode = 1 Compressibility Factor
Kode = 2 Enthalpy Departure, $\Delta$H/Tc, J/mol-K
Kode = 3 Entropy Deaparture, $\Delta$S, J/mol-K
Kode = 4 Fugacity Coefficient

Four separate sheets in Excel are set up and the same function procedure is called from each sheet.

## The Spreadsheet

Figures 2 to 5 show the spreadsheets used for calculating the compressibility, isothermal enthalpy departure, the isothermal pressure departure and fugacity coefficient as a function of reduced pressure and temperature.

The function procedure RKfunc is invoked at each table entry as:

```
= RKfunc (Tc, Pc, Tr, Pr, Kode)
```

For the entry at Pr = 5 and Tr = 1.2 the entry would be

```
= RKfunc ($B$5,$B$6, F$9,  $A35,  $B$7)
```

```
where $B$5 = Location of Tc
      $B$6 = Location of Pc
       F$9 = Location of Tr
      $A35 = Location of Pr
      $B$7 = Location of Kode
```

## Comparison to Charts

Table 1 compares the results of the VBA/spreadsheet program (Figure 6) to that given in classical charts (5). Since the classical charts are based on the theory of corresponding states and not on the Redlich Kwong Equation of State, the differences are not surprising (6).

| | Tr | Pr | VBA (Spreadsheet) | Chart |
|---|---|---|---|---|
| Compressibility | 1.40 | 1.51 | .82 | .84 |
| Enthalpy Departure J/(mol-K) | 1.23 | 1.37 | 9.65 | 10.66 |
| Entropy Departure J/(mol K) | 1.23 | 1.37 | 5.66 | 6.23 |
| Fugacity Coefficient | 1.70 | 1.49 | .92 | .94 |

**Table 1**
**VBA Comparison to Charts**

## Conclusions

The overall programming effort for this exercise in Matlab and VBA is about the same, though the spreadsheet's main program is somewhat simpler. Which programming language is used in practice, however, is a function of program availability, cost and user training.

## References


1. Shacham, M, Brauner, N. and Cutlip, M., "An Exercise for Practicing Programming in the ChE Curriculum", *Chemical Engineering Education* vol 37, No. 2, Spring 2003 p. 148

2. MATLAB is a trademark of The Math Works, Inc http://www.mathworks.com

3. Walkenbach, John, *Microsoft Excel 2000 Power Programming with VBA* IDG Books Worldwide, Inc., Foster City, Ca 1999

4. Misovich, M. J. “Making Phase Equilibrium More User-Friendly”, *Chemical Engineering Education*, Fall 2002 36, 4 p 284

5. Hougen, O. A. and Watson, K. M. *Chemical Process Principles Part Two Thermodynamics*, John Wiley (New York) 1948 pages 490,496,499,621

6. Peress, J. “Working With Non-Ideal Gases, *Chemical Engineering Progress*, March 2003, p 39.

```
Option Explicit
Public Function RKfunc(Tc, Pc, Tr, Pr, Kode)
'  Pc in pascals
'  Tc in kelvins
'  Tr  Reduced Temperature
'  Pr  Reduced Pressure

'  Kode = 1   z
'  Kode = 2   Hdep
'  Kode = 3   Sdep
'  Kode = 4   fcoeff

Dim a, bb, Asqr, B, R, q, f, g, C As Single

Dim D, E1, E, z As Single
Dim psii, zv(3) As Single

Dim P, T, V  As Single
Dim Hdep, Sdep, f_coeff

Dim rr As Single
Dim E1x, E1y

Dim zz, zzz, s, ss As Single

' R in J/mole/K
R = 8.3143

a = 0.42747 * R ^ 2 * Tc ^ (5 / 2) / Pc
bb = 0.08664 * R * Tc / Pc

Asqr = 0.42747 * Pr / (Tr ^ 2.5)
B = 0.08664 * Pr / Tr
rr = Asqr * B
q = B ^ 2 + B - Asqr
f = (-3 * q - 1) / 3
g = (-27 * rr - 9 * q - 2) / 27
C = (f / 3) ^ 3 + (g / 2) ^ 2

If C > 0 Then

D = ((-g / 2 + Sqr(C)) ^ (1 / 3))
E1 = (-g / 2 - Sqr(C))

E = ((Sgn(E1) * (Abs(E1)) ^ (1 / 3)))

z = (D + E + 1 / 3)

Else

psii = (Application.Acos(Sqr((g ^ 2 / 4) / (-f ^ 3 / 27))))

zv(1) = (2 * Sqr(-f / 3) * Cos((psii / 3)) + 1 / 3)
zv(2) = (2 * Sqr(-f / 3) * Cos((psii / 3) + 2 * 3.14159 * 1 / 3) + 1 / 3)
zv(3) = (2 * Sqr(-f / 3) * Cos((psii / 3) + 2 * 3.14159 * 2 / 3) + 1 / 3)
```

```
z = Application.Max(zv(1), zv(2), zv(3))

End If

P = Pr * Pc
T = Tr * Tc
V = z * R * T / P
Hdep = (3 * a / (2 * bb * R * T ^ 1.5)) * Log(1 + bb / V) - (z - 1)

'Put in Standard Form
Hdep = Hdep * R * T
Hdep = Hdep / Tc


Sdep = (a / (2 * bb * R * T ^ 1.5)) * Log((1 + bb / V)) - Log(z - P * bb / (R *
T))
'Put in Standard Form
Sdep = Sdep * R

f_coeff = Exp(z - 1 - Log(z * (1 - bb / V)) - a / (bb * R * T ^ 1.5) * Log(1 +
bb / V))

If Kode = 1 Then RKfunc = z
If Kode = 2 Then RKfunc = Hdep
If Kode = 3 Then RKfunc = Sdep
If Kode = 4 Then RKfunc = f_coeff


End Function
```

**Figure 1**
**Public Function RKfunc**

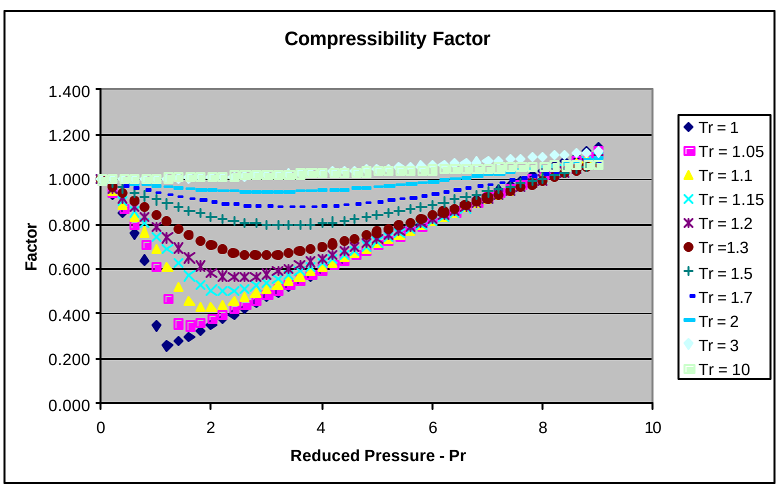
An Exercise - CEE Spring 2003 p 148

Water

***Figure 2 Compressibility Factor***

Tc 647.4 Kelvins
Pc 22119248 Pascals
Kode 1

| Pr | Tr = 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.3 | 1.5 | 1.7 | 2 | 3 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0.2 | 0.929 | 0.9387 | 0.9469 | 0.9538 | 0.9596 | 0.9688 | 0.9807 | 0.9877 | 0.9937 | 1.0004 | 1.0015 |
| 0.4 | 0.849 | 0.8721 | 0.8905 | 0.9056 | 0.918 | 0.9372 | 0.9616 | 0.9758 | 0.9878 | 1.0008 | 1.0029 |
| 0.6 | 0.757 | 0.7983 | 0.83 | 0.8549 | 0.875 | 0.9054 | 0.9429 | 0.9643 | 0.9822 | 1.0015 | 1.0044 |
| 0.8 | 0.639 | 0.7137 | 0.7641 | 0.8015 | 0.8307 | 0.8735 | 0.9248 | 0.9534 | 0.977 | 1.0022 | 1.0059 |
| 1 | 0.347 | 0.6101 | 0.6911 | 0.7451 | 0.7852 | 0.8418 | 0.9072 | 0.943 | 0.9722 | 1.0031 | 1.0073 |
| 1.2 | 0.258 | 0.4666 | 0.6093 | 0.6859 | 0.7389 | 0.8105 | 0.8905 | 0.9332 | 0.9678 | 1.0041 | 1.0088 |
| 1.4 | 0.277 | 0.3575 | 0.5219 | 0.6256 | 0.693 | 0.7803 | 0.8746 | 0.9241 | 0.9639 | 1.0052 | 1.0103 |
| 1.6 | 0.300 | 0.3497 | 0.4559 | 0.5697 | 0.6496 | 0.7518 | 0.8598 | 0.9157 | 0.9604 | 1.0064 | 1.0118 |
| 1.8 | 0.324 | 0.3618 | 0.4318 | 0.5281 | 0.6123 | 0.7259 | 0.8462 | 0.9081 | 0.9573 | 1.0078 | 1.0133 |
| 2 | 0.349 | 0.3796 | 0.4313 | 0.5061 | 0.5846 | 0.7035 | 0.834 | 0.9014 | 0.9547 | 1.0093 | 1.0148 |
| 2.2 | 0.374 | 0.3998 | 0.4409 | 0.4998 | 0.568 | 0.6856 | 0.8234 | 0.8954 | 0.9525 | 1.0109 | 1.0162 |
| 2.4 | 0.399 | 0.4211 | 0.4553 | 0.5033 | 0.5612 | 0.6725 | 0.8145 | 0.8904 | 0.9508 | 1.0126 | 1.0177 |
| 2.6 | 0.423 | 0.443 | 0.4722 | 0.5124 | 0.5618 | 0.6642 | 0.8073 | 0.8863 | 0.9496 | 1.0145 | 1.0192 |
| 2.8 | 0.448 | 0.4651 | 0.4905 | 0.525 | 0.5675 | 0.6605 | 0.8019 | 0.8831 | 0.9489 | 1.0165 | 1.0207 |
| 3 | 0.472 | 0.4874 | 0.5097 | 0.5397 | 0.5767 | 0.6606 | 0.7983 | 0.8808 | 0.9486 | 1.0185 | 1.0222 |
| 3.2 | 0.497 | 0.5098 | 0.5294 | 0.5557 | 0.5883 | 0.6639 | 0.7963 | 0.8794 | 0.9487 | 1.0207 | 1.0237 |
| 3.4 | 0.521 | 0.5321 | 0.5494 | 0.5727 | 0.6016 | 0.6698 | 0.796 | 0.879 | 0.9494 | 1.023 | 1.0252 |
| 3.6 | 0.544 | 0.5544 | 0.5697 | 0.5904 | 0.6161 | 0.6777 | 0.7973 | 0.8794 | 0.9504 | 1.0254 | 1.0267 |
| 3.8 | 0.568 | 0.5766 | 0.5901 | 0.6085 | 0.6315 | 0.6872 | 0.7999 | 0.8806 | 0.9519 | 1.0279 | 1.0282 |
| 4 | 0.592 | 0.5988 | 0.6105 | 0.6269 | 0.6475 | 0.698 | 0.8037 | 0.8827 | 0.9538 | 1.0305 | 1.0297 |
| 4.2 | 0.615 | 0.6208 | 0.631 | 0.6455 | 0.6639 | 0.7097 | 0.8088 | 0.8855 | 0.9561 | 1.0333 | 1.0313 |
| 4.4 | 0.638 | 0.6428 | 0.6515 | 0.6643 | 0.6808 | 0.7223 | 0.8148 | 0.889 | 0.9588 | 1.0361 | 1.0328 |
| 4.6 | 0.661 | 0.6646 | 0.672 | 0.6831 | 0.6978 | 0.7355 | 0.8217 | 0.8931 | 0.9619 | 1.039 | 1.0343 |
| 4.8 | 0.684 | 0.6864 | 0.6924 | 0.7021 | 0.7151 | 0.7492 | 0.8293 | 0.8979 | 0.9654 | 1.042 | 1.0358 |
| 5 | 0.707 | 0.708 | 0.7128 | 0.721 | 0.7326 | 0.7633 | 0.8376 | 0.9032 | 0.9692 | 1.0451 | 1.0373 |
| 5.2 | 0.730 | 0.7296 | 0.7331 | 0.74 | 0.7501 | 0.7777 | 0.8466 | 0.909 | 0.9733 | 1.0482 | 1.0388 |
| 5.4 | 0.752 | 0.7511 | 0.7534 | 0.759 | 0.7677 | 0.7924 | 0.856 | 0.9154 | 0.9777 | 1.0515 | 1.0404 |
| 5.6 | 0.775 | 0.7724 | 0.7736 | 0.778 | 0.7854 | 0.8074 | 0.8659 | 0.9221 | 0.9825 | 1.0549 | 1.0419 |
| 5.8 | 0.797 | 0.7937 | 0.7937 | 0.797 | 0.8031 | 0.8225 | 0.8762 | 0.9293 | 0.9875 | 1.0583 | 1.0434 |
| 6 | 0.820 | 0.8149 | 0.8138 | 0.8159 | 0.8209 | 0.8377 | 0.8868 | 0.9368 | 0.9927 | 1.0618 | 1.045 |
| 6.2 | 0.842 | 0.836 | 0.8338 | 0.8348 | 0.8386 | 0.8531 | 0.8978 | 0.9447 | 0.9983 | 1.0654 | 1.0465 |
| 6.4 | 0.864 | 0.857 | 0.8538 | 0.8537 | 0.8564 | 0.8686 | 0.909 | 0.9528 | 1.004 | 1.0691 | 1.048 |
| 6.6 | 0.886 | 0.878 | 0.8737 | 0.8726 | 0.8742 | 0.8842 | 0.9204 | 0.9613 | 1.01 | 1.0728 | 1.0496 |
| 6.8 | 0.908 | 0.8989 | 0.8935 | 0.8914 | 0.8919 | 0.8998 | 0.9321 | 0.97 | 1.0162 | 1.0766 | 1.0511 |
| 7 | 0.929 | 0.9196 | 0.9133 | 0.9101 | 0.9097 | 0.9155 | 0.9439 | 0.9789 | 1.0226 | 1.0805 | 1.0526 |
| 7.2 | 0.951 | 0.9404 | 0.933 | 0.9288 | 0.9274 | 0.9312 | 0.9559 | 0.988 | 1.0292 | 1.0845 | 1.0542 |
| 7.4 | 0.973 | 0.961 | 0.9527 | 0.9475 | 0.9451 | 0.9469 | 0.968 | 0.9973 | 1.0359 | 1.0885 | 1.0557 |
| 7.6 | 0.994 | 0.9816 | 0.9723 | 0.9661 | 0.9627 | 0.9627 | 0.9803 | 1.0069 | 1.0428 | 1.0926 | 1.0573 |
| 7.8 | 1.016 | 1.0021 | 0.9918 | 0.9847 | 0.9804 | 0.9785 | 0.9926 | 1.0164 | 1.0499 | 1.0967 | 1.0588 |
| 8 | 1.037 | 1.0225 | 1.0113 | 1.0033 | 0.998 | 0.9943 | 1.0051 | 1.0276 | 1.0571 | 1.1009 | 1.0604 |
| 8.2 | 1.059 | 1.0429 | 1.0307 | 1.0218 | 1.0156 | 1.0101 | 1.0177 | 1.0363 | 1.0644 | 1.1052 | 1.0619 |
| 8.4 | 1.080 | 1.0632 | 1.0501 | 1.0402 | 1.0331 | 1.0259 | 1.0303 | 1.0462 | 1.0719 | 1.1095 | 1.0635 |
| 8.6 | 1.101 | 1.0834 | 1.0694 | 1.0586 | 1.0507 | 1.0417 | 1.043 | 1.0564 | 1.0795 | 1.1138 | 1.065 |
| 8.8 | 1.122 | 1.1036 | 1.0887 | 1.077 | 1.0682 | 1.0576 | 1.0557 | 1.0667 | 1.0871 | 1.1183 | 1.0666 |
| 9 | 1.143 | 1.1238 | 1.1079 | 1.0953 | 1.0856 | 1.0734 | 1.0685 | 1.077 | 1.0949 | 1.1227 | 1.0681 |



An Exercise - CEE Spring 2003 p 148

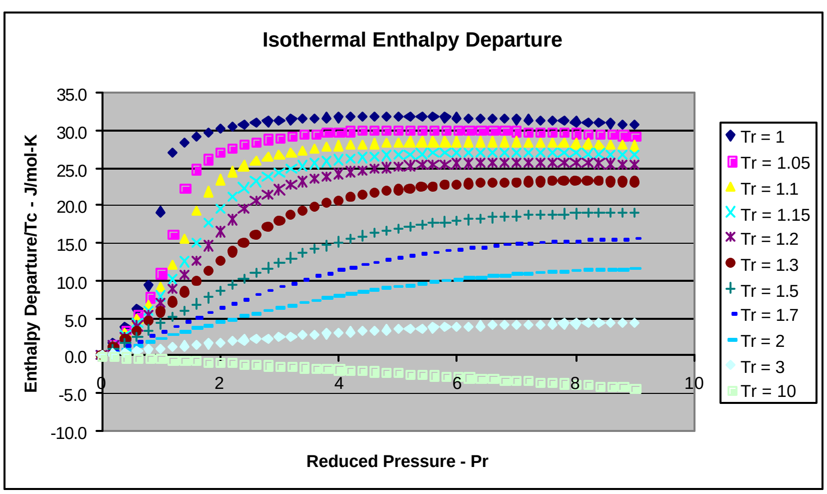
Water

***Figure 3 Enthalpy Departure***

| | | |
|---|---|---|
| Tc | 647.4 | Kelvins |
| Pc | 22119248 | Pascals |
| Kode | 2 | |

<==================================Tr================================>

| Pr | 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.3 | 1.5 | 1.7 | 2 | 3 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1E-08 | 0.000 | 5E-08 | 4.6E-08 | 4E-08 | 4E-08 | 3.6E-08 | 3E-08 | 2E-08 | 2E-08 | 1E-08 | 2E-09 |
| 0.2 | 1.731 | 1.582 | 1.45324 | 1.3408 | 1.2416 | 1.07499 | 0.8297 | 0.6585 | 0.4822 | 0.1956 | -0.088 |
| 0.4 | 3.717 | 3.3471 | 3.0407 | 2.7812 | 2.558 | 2.19228 | 1.6717 | 1.318 | 0.9598 | 0.3867 | -0.177 |
| 0.6 | 6.113 | 5.37348 | 4.8037 | 4.3437 | 3.961 | 3.35471 | 2.5249 | 1.9774 | 1.4322 | 0.5731 | -0.265 |
| 0.8 | 9.343 | 7.81172 | 6.80689 | 6.0581 | 5.4647 | 4.56446 | 3.3876 | 2.6351 | 1.8987 | 0.7548 | -0.354 |
| 1 | 19.159 | 11.0216 | 9.1573 | 7.9639 | 7.0842 | 5.82231 | 4.2577 | 3.2897 | 2.3586 | 0.9318 | -0.444 |
| 1.2 | 27.013 | 16.2124 | 12.0339 | 10.108 | 8.8325 | 7.12636 | 5.1321 | 3.9394 | 2.8111 | 1.104 | -0.534 |
| 1.4 | 28.376 | 22.4238 | 15.6172 | 12.52 | 10.711 | 8.47028 | 6.0071 | 4.5824 | 3.2555 | 1.2714 | -0.624 |
| 1.6 | 29.202 | 24.9087 | 19.2964 | 15.128 | 12.69 | 9.84102 | 6.8782 | 5.2166 | 3.6912 | 1.4341 | -0.715 |
| 1.8 | 29.780 | 26.2264 | 21.8563 | 17.628 | 14.681 | 11.2168 | 7.74 | 5.8399 | 4.1173 | 1.5919 | -0.806 |
| 2 | 30.215 | 27.0932 | 23.4669 | 19.677 | 16.545 | 12.5675 | 8.5867 | 6.4503 | 4.5334 | 1.7449 | -0.897 |
| 2.2 | 30.554 | 27.7214 | 24.5597 | 21.215 | 18.162 | 13.8578 | 9.4121 | 7.0456 | 4.9386 | 1.8931 | -0.989 |
| 2.4 | 30.824 | 28.2016 | 25.3555 | 22.362 | 19.493 | 15.0556 | 10.21 | 7.6238 | 5.3326 | 2.0365 | -1.081 |
| 2.6 | 31.042 | 28.5807 | 25.9637 | 23.237 | 20.566 | 16.1386 | 10.975 | 8.1831 | 5.7148 | 2.1751 | -1.173 |
| 2.8 | 31.220 | 28.8864 | 26.444 | 23.924 | 21.431 | 17.0982 | 11.701 | 8.7219 | 6.0847 | 2.3089 | -1.266 |
| 3 | 31.366 | 29.1362 | 26.8318 | 24.474 | 22.135 | 17.937 | 12.386 | 9.2386 | 6.442 | 2.438 | -1.359 |
| 3.2 | 31.484 | 29.3422 | 27.1499 | 24.924 | 22.714 | 18.6649 | 13.026 | 9.7322 | 6.7863 | 2.5623 | -1.452 |
| 3.4 | 31.581 | 29.5128 | 27.4137 | 25.296 | 23.196 | 19.2947 | 13.622 | 10.202 | 7.1175 | 2.6819 | -1.546 |
| 3.6 | 31.658 | 29.6542 | 27.6339 | 25.607 | 23.601 | 19.8397 | 14.172 | 10.647 | 7.4354 | 2.7968 | -1.64 |
| 3.8 | 31.720 | 29.771 | 27.8184 | 25.869 | 23.944 | 20.3119 | 14.677 | 11.068 | 7.7399 | 2.9071 | -1.734 |
| 4 | 31.766 | 29.8671 | 27.9734 | 26.09 | 24.235 | 20.7221 | 15.143 | 11.465 | 8.031 | 3.0128 | -1.829 |
| 4.2 | 31.801 | 29.9452 | 28.1032 | 26.279 | 24.484 | 21.0791 | 15.564 | 11.837 | 8.3087 | 3.1139 | -1.924 |
| 4.4 | 31.824 | 30.0078 | 28.2116 | 26.438 | 24.698 | 21.3906 | 15.95 | 12.185 | 8.5732 | 3.2106 | -2.02 |
| 4.6 | 31.838 | 30.0566 | 28.3014 | 26.574 | 24.881 | 21.6628 | 16.301 | 12.511 | 8.8245 | 3.3028 | -2.115 |
| 4.8 | 31.842 | 30.0934 | 28.375 | 26.688 | 25.039 | 21.9008 | 16.619 | 12.815 | 9.063 | 3.3906 | -2.211 |
| 5 | 31.839 | 30.1193 | 28.4343 | 26.784 | 25.173 | 22.1091 | 16.908 | 13.097 | 9.2889 | 3.4741 | -2.307 |
| 5.2 | 31.829 | 30.1356 | 28.4809 | 26.864 | 25.288 | 22.2913 | 17.169 | 13.359 | 9.5024 | 3.5534 | -2.404 |
| 5.4 | 31.811 | 30.1432 | 28.5163 | 26.93 | 25.385 | 22.4505 | 17.405 | 13.602 | 9.7038 | 3.6284 | -2.501 |
| 5.6 | 31.788 | 30.1428 | 28.5415 | 26.982 | 25.467 | 22.5892 | 17.619 | 13.826 | 9.8936 | 3.6993 | -2.598 |
| 5.8 | 31.760 | 30.1353 | 28.5575 | 27.024 | 25.535 | 22.7097 | 17.811 | 14.033 | 10.072 | 3.7661 | -2.695 |
| 6 | 31.726 | 30.1213 | 28.5652 | 27.055 | 25.591 | 22.8138 | 17.984 | 14.224 | 10.239 | 3.8289 | -2.793 |
| 6.2 | 31.687 | 30.1012 | 28.5653 | 27.077 | 25.636 | 22.9032 | 18.14 | 14.399 | 10.396 | 3.8877 | -2.891 |
| 6.4 | 31.645 | 30.0755 | 28.5585 | 27.09 | 25.67 | 22.9794 | 18.279 | 14.56 | 10.543 | 3.9427 | -2.99 |
| 6.6 | 31.598 | 30.0448 | 28.5453 | 27.096 | 25.695 | 23.0435 | 18.403 | 14.707 | 10.679 | 3.9938 | -3.088 |
| 6.8 | 31.547 | 30.0093 | 28.5263 | 27.094 | 25.712 | 23.0967 | 18.514 | 14.842 | 10.806 | 4.0413 | -3.187 |
| 7 | 31.493 | 29.9694 | 28.5018 | 27.086 | 25.721 | 23.1398 | 18.611 | 14.964 | 10.924 | 4.085 | -3.287 |
| 7.2 | 31.436 | 29.9255 | 28.4724 | 27.072 | 25.723 | 23.1737 | 18.697 | 15.075 | 11.033 | 4.1252 | -3.386 |
| 7.4 | 31.375 | 29.8777 | 28.4383 | 27.053 | 25.718 | 23.1991 | 18.772 | 15.174 | 11.134 | 4.1618 | -3.486 |
| 7.6 | 31.312 | 29.8263 | 28.3999 | 27.028 | 25.708 | 23.2168 | 18.838 | 15.263 | 11.227 | 4.195 | -3.586 |
| 7.8 | 31.246 | 29.7716 | 28.3574 | 26.998 | 25.691 | 23.2273 | 18.893 | 15.347 | 11.311 | 4.2247 | -3.686 |
| 8 | 31.177 | 29.7138 | 28.3112 | 26.964 | 25.67 | 23.2311 | 18.94 | 15.38 | 11.389 | 4.2512 | -3.787 |
| 8.2 | 31.106 | 29.653 | 28.2614 | 26.926 | 25.644 | 23.2286 | 18.979 | 15.475 | 11.459 | 4.2744 | -3.888 |
| 8.4 | 31.033 | 29.5894 | 28.2083 | 26.884 | 25.613 | 23.2204 | 19.01 | 15.537 | 11.522 | 4.2944 | -3.989 |
| 8.6 | 30.957 | 29.5233 | 28.152 | 26.838 | 25.578 | 23.2069 | 19.034 | 15.582 | 11.579 | 4.3114 | -4.09 |
| 8.8 | 30.880 | 29.4546 | 28.0928 | 26.789 | 25.538 | 23.1882 | 19.052 | 15.622 | 11.63 | 4.3252 | -4.192 |
| 9 | 30.800 | 29.3836 | 28.0309 | 26.736 | 25.496 | 23.1649 | 19.063 | 15.656 | 11.674 | 4.3361 | -4.294 |



An Exercise - CEE Spring 2003 p 148

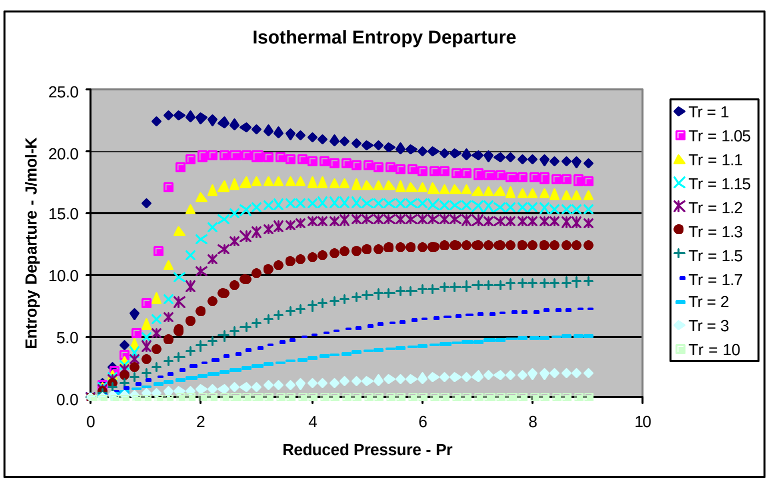
Water

***Figure 4 Entropy Departure***

Tc 647.4 Kelvins
Pc 22119248 Pascals
Kode 3

<==================================Tr================================>

| Pr | 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.3 | 1.5 | 1.7 | 2 | 3 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.000 | 2E-08 | 2E-08 | 2E-08 | 2E-08 | 1E-08 | 1E-08 | 9E-09 | 7E-09 | 4E-09 | 8E-10 |
| 0.2 | 1.151 | 1.006 | 0.8861 | 0.7861 | 0.7016 | 0.5681 | 0.3917 | 0.2842 | 0.1882 | 0.0679 | 0.0034 |
| 0.4 | 2.527 | 2.1666 | 1.8813 | 1.6505 | 1.4604 | 1.1672 | 0.7928 | 0.5707 | 0.3755 | 0.1348 | 0.0067 |
| 0.6 | 4.274 | 3.5518 | 3.0212 | 2.6119 | 2.286 | 1.7997 | 1.2027 | 0.8587 | 0.5616 | 0.2007 | 0.01 |
| 0.8 | 6.795 | 5.2979 | 4.3618 | 3.6955 | 3.19 | 2.4678 | 1.6206 | 1.1478 | 0.7463 | 0.2656 | 0.0133 |
| 1 | 15.775 | 7.7352 | 5.9969 | 4.9345 | 4.185 | 3.1723 | 2.0454 | 1.437 | 0.9293 | 0.3294 | 0.0165 |
| 1.2 | 22.507 | 11.989 | 8.0847 | 6.3691 | 5.2823 | 3.9125 | 2.4752 | 1.7255 | 1.1102 | 0.3922 | 0.0198 |
| 1.4 | 22.930 | 17.136 | 10.787 | 8.0267 | 6.4843 | 4.6844 | 2.9082 | 2.0124 | 1.2886 | 0.4539 | 0.023 |
| 1.6 | 22.965 | 18.781 | 13.561 | 9.8471 | 7.7684 | 5.4792 | 3.3415 | 2.2966 | 1.4644 | 0.5146 | 0.0261 |
| 1.8 | 22.869 | 19.404 | 15.341 | 11.578 | 9.0659 | 6.2818 | 3.7721 | 2.577 | 1.6372 | 0.5742 | 0.0293 |
| 2 | 22.723 | 19.678 | 16.306 | 12.935 | 10.267 | 7.0707 | 4.1965 | 2.8526 | 1.8066 | 0.6326 | 0.0324 |
| 2.2 | 22.555 | 19.793 | 16.852 | 13.879 | 11.278 | 7.8211 | 4.611 | 3.1223 | 1.9725 | 0.69 | 0.0355 |
| 2.4 | 22.381 | 19.823 | 17.176 | 14.515 | 12.072 | 8.51 | 5.0119 | 3.3849 | 2.1345 | 0.7463 | 0.0386 |
| 2.6 | 22.207 | 19.806 | 17.372 | 14.948 | 12.674 | 9.1222 | 5.3958 | 3.6396 | 2.2924 | 0.8015 | 0.0417 |
| 2.8 | 22.038 | 19.761 | 17.489 | 15.248 | 13.126 | 9.6521 | 5.7597 | 3.8854 | 2.4461 | 0.8557 | 0.0447 |
| 3 | 21.874 | 19.698 | 17.555 | 15.459 | 13.467 | 10.102 | 6.1014 | 4.1216 | 2.5953 | 0.9087 | 0.0477 |
| 3.2 | 21.716 | 19.625 | 17.586 | 15.607 | 13.726 | 10.481 | 6.4196 | 4.3476 | 2.7399 | 0.9607 | 0.0507 |
| 3.4 | 21.564 | 19.546 | 17.593 | 15.711 | 13.923 | 10.797 | 6.7136 | 4.563 | 2.8798 | 1.0115 | 0.0536 |
| 3.6 | 21.420 | 19.464 | 17.584 | 15.782 | 14.075 | 11.062 | 6.9836 | 4.7675 | 3.0149 | 1.0613 | 0.0566 |
| 3.8 | 21.281 | 19.38 | 17.563 | 15.83 | 14.191 | 11.282 | 7.2302 | 4.9611 | 3.1451 | 1.1101 | 0.0595 |
| 4 | 21.149 | 19.296 | 17.534 | 15.859 | 14.28 | 11.466 | 7.4559 | 5.1438 | 3.2705 | 1.1578 | 0.0624 |
| 4.2 | 21.022 | 19.212 | 17.498 | 15.875 | 14.348 | 11.621 | 7.6582 | 5.3157 | 3.3911 | 1.2044 | 0.0653 |
| 4.4 | 20.901 | 19.129 | 17.457 | 15.881 | 14.399 | 11.751 | 7.8424 | 5.4771 | 3.5069 | 1.25 | 0.0681 |
| 4.6 | 20.785 | 19.047 | 17.414 | 15.878 | 14.437 | 11.86 | 8.009 | 5.6284 | 3.6179 | 1.2946 | 0.0709 |
| 4.8 | 20.674 | 18.967 | 17.368 | 15.869 | 14.465 | 11.952 | 8.1595 | 5.7699 | 3.7243 | 1.3382 | 0.0737 |
| 5 | 20.568 | 18.889 | 17.321 | 15.854 | 14.483 | 12.029 | 8.2954 | 5.9022 | 3.8261 | 1.3808 | 0.0765 |
| 5.2 | 20.465 | 18.813 | 17.273 | 15.836 | 14.494 | 12.094 | 8.4181 | 6.0257 | 3.9235 | 1.4224 | 0.0793 |
| 5.4 | 20.367 | 18.739 | 17.225 | 15.814 | 14.5 | 12.149 | 8.529 | 6.141 | 4.0165 | 1.4631 | 0.082 |
| 5.6 | 20.272 | 18.666 | 17.176 | 15.79 | 14.5 | 12.195 | 8.6292 | 6.2484 | 4.1053 | 1.5028 | 0.0847 |
| 5.8 | 20.181 | 18.596 | 17.128 | 15.764 | 14.497 | 12.234 | 8.7198 | 6.3486 | 4.1901 | 1.5416 | 0.0874 |
| 6 | 20.093 | 18.527 | 17.079 | 15.737 | 14.491 | 12.266 | 8.8018 | 6.4419 | 4.271 | 1.5794 | 0.0901 |
| 6.2 | 20.009 | 18.461 | 17.031 | 15.708 | 14.481 | 12.293 | 8.876 | 6.5289 | 4.3482 | 1.6164 | 0.0928 |
| 6.4 | 19.927 | 18.396 | 16.984 | 15.679 | 14.47 | 12.315 | 8.9433 | 6.6099 | 4.4217 | 1.6524 | 0.0954 |
| 6.6 | 19.848 | 18.333 | 16.937 | 15.648 | 14.456 | 12.333 | 9.0043 | 6.6855 | 4.4918 | 1.6876 | 0.098 |
| 6.8 | 19.772 | 18.271 | 16.891 | 15.618 | 14.441 | 12.347 | 9.0596 | 6.7559 | 4.5586 | 1.722 | 0.1006 |
| 7 | 19.698 | 18.211 | 16.846 | 15.587 | 14.425 | 12.357 | 9.1099 | 6.8215 | 4.6222 | 1.7555 | 0.1032 |
| 7.2 | 19.627 | 18.153 | 16.801 | 15.556 | 14.407 | 12.366 | 9.1555 | 6.8829 | 4.6828 | 1.7882 | 0.1058 |
| 7.4 | 19.558 | 18.096 | 16.757 | 15.525 | 14.389 | 12.371 | 9.1969 | 6.9399 | 4.7405 | 1.8201 | 0.1083 |
| 7.6 | 19.491 | 18.041 | 16.713 | 15.494 | 14.37 | 12.375 | 9.2345 | 6.9927 | 4.7956 | 1.8512 | 0.1108 |
| 7.8 | 19.426 | 17.987 | 16.671 | 15.463 | 14.35 | 12.377 | 9.2687 | 7.0442 | 4.848 | 1.8816 | 0.1133 |
| 8 | 19.363 | 17.934 | 16.629 | 15.432 | 14.33 | 12.377 | 9.2998 | 7.0684 | 4.8979 | 1.9112 | 0.1158 |
| 8.2 | 19.301 | 17.883 | 16.588 | 15.401 | 14.309 | 12.375 | 9.328 | 7.1308 | 4.9455 | 1.9401 | 0.1183 |
| 8.4 | 19.242 | 17.833 | 16.548 | 15.371 | 14.288 | 12.372 | 9.3536 | 7.175 | 4.9908 | 1.9683 | 0.1207 |
| 8.6 | 19.184 | 17.785 | 16.509 | 15.34 | 14.267 | 12.369 | 9.3768 | 7.2117 | 5.034 | 1.9957 | 0.1231 |
| 8.8 | 19.128 | 17.737 | 16.47 | 15.31 | 14.246 | 12.364 | 9.3979 | 7.2469 | 5.0752 | 2.0225 | 0.1255 |
| 9 | 19.073 | 17.691 | 16.432 | 15.281 | 14.225 | 12.358 | 9.417 | 7.2805 | 5.1144 | 2.0487 | 0.1279 |



An Exercise - CEE Spring 2003 p 148

Water

***Figure 5 Fugacity Coeffficient***

Tc 647.4 Kelvins
Pc 22119248 Pascals
Kode 4

<================================Tr==================================>

| Pr | 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.3 | 1.5 | 1.7 | 2 | 3 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0.2 | 0.933 | 0.9416 | 0.949 | 0.9553 | 0.96073 | 0.9693 | 0.98077 | 0.9877 | 0.9937 | 1.0003 | 1.001 |
| 0.4 | 0.867 | 0.8844 | 0.8992 | 0.9118 | 0.92244 | 0.9395 | 0.96205 | 0.9757 | 0.9875 | 1.0007 | 1.003 |
| 0.6 | 0.802 | 0.8283 | 0.8506 | 0.8692 | 0.88509 | 0.9104 | 0.94384 | 0.964 | 0.9816 | 1.0012 | 1.004 |
| 0.8 | 0.736 | 0.7729 | 0.8028 | 0.8277 | 0.8487 | 0.8821 | 0.92616 | 0.9528 | 0.9759 | 1.0017 | 1.006 |
| 1 | 0.666 | 0.7174 | 0.7558 | 0.7871 | 0.81327 | 0.8546 | 0.90902 | 0.9418 | 0.9704 | 1.0023 | 1.007 |
| 1.2 | 0.582 | 0.6603 | 0.7094 | 0.7474 | 0.77885 | 0.828 | 0.89244 | 0.9313 | 0.9651 | 1.0029 | 1.009 |
| 1.4 | 0.519 | 0.6019 | 0.6635 | 0.7089 | 0.74554 | 0.8023 | 0.87644 | 0.9211 | 0.96 | 1.0036 | 1.01 |
| 1.6 | 0.472 | 0.5519 | 0.6195 | 0.6718 | 0.71354 | 0.7777 | 0.86104 | 0.9113 | 0.9552 | 1.0044 | 1.012 |
| 1.8 | 0.436 | 0.5116 | 0.58 | 0.6369 | 0.68316 | 0.7541 | 0.84626 | 0.9019 | 0.9506 | 1.0052 | 1.013 |
| 2 | 0.406 | 0.4787 | 0.5463 | 0.6053 | 0.65481 | 0.7318 | 0.83212 | 0.8929 | 0.9462 | 1.0062 | 1.015 |
| 2.2 | 0.382 | 0.4516 | 0.5176 | 0.5772 | 0.62885 | 0.7108 | 0.81865 | 0.8843 | 0.942 | 1.0071 | 1.016 |
| 2.4 | 0.362 | 0.4291 | 0.4934 | 0.5527 | 0.60544 | 0.6912 | 0.80585 | 0.8761 | 0.938 | 1.0082 | 1.018 |
| 2.6 | 0.346 | 0.41 | 0.4726 | 0.5313 | 0.58453 | 0.673 | 0.79374 | 0.8683 | 0.9343 | 1.0092 | 1.019 |
| 2.8 | 0.331 | 0.3937 | 0.4548 | 0.5127 | 0.56596 | 0.6564 | 0.78232 | 0.8609 | 0.9308 | 1.0104 | 1.021 |
| 3 | 0.319 | 0.3797 | 0.4394 | 0.4964 | 0.54948 | 0.6412 | 0.77159 | 0.8539 | 0.9275 | 1.0116 | 1.022 |
| 3.2 | 0.309 | 0.3676 | 0.4259 | 0.4821 | 0.53487 | 0.6273 | 0.76156 | 0.8473 | 0.9244 | 1.0129 | 1.024 |
| 3.4 | 0.300 | 0.3571 | 0.4142 | 0.4695 | 0.52189 | 0.6148 | 0.7522 | 0.8412 | 0.9216 | 1.0142 | 1.025 |
| 3.6 | 0.292 | 0.3479 | 0.4039 | 0.4584 | 0.51034 | 0.6034 | 0.7435 | 0.8354 | 0.9189 | 1.0156 | 1.027 |
| 3.8 | 0.285 | 0.3398 | 0.3948 | 0.4486 | 0.50006 | 0.5932 | 0.73545 | 0.83 | 0.9165 | 1.0171 | 1.028 |
| 4 | 0.279 | 0.3327 | 0.3868 | 0.4399 | 0.4909 | 0.5839 | 0.72801 | 0.8249 | 0.9143 | 1.0186 | 1.03 |
| 4.2 | 0.274 | 0.3265 | 0.3797 | 0.4322 | 0.48272 | 0.5755 | 0.72115 | 0.8203 | 0.9123 | 1.0202 | 1.031 |
| 4.4 | 0.269 | 0.3209 | 0.3734 | 0.4253 | 0.47541 | 0.5679 | 0.71486 | 0.816 | 0.9105 | 1.0219 | 1.033 |
| 4.6 | 0.265 | 0.316 | 0.3679 | 0.4191 | 0.46889 | 0.5611 | 0.70911 | 0.812 | 0.9089 | 1.0236 | 1.035 |
| 4.8 | 0.261 | 0.3117 | 0.3629 | 0.4137 | 0.46307 | 0.555 | 0.70386 | 0.8084 | 0.9075 | 1.0253 | 1.036 |
| 5 | 0.258 | 0.3078 | 0.3585 | 0.4089 | 0.45788 | 0.5495 | 0.69909 | 0.8052 | 0.9063 | 1.0272 | 1.038 |
| 5.2 | 0.255 | 0.3045 | 0.3547 | 0.4046 | 0.45325 | 0.5446 | 0.69477 | 0.8022 | 0.9052 | 1.029 | 1.039 |
| 5.4 | 0.252 | 0.3015 | 0.3512 | 0.4007 | 0.44915 | 0.5402 | 0.69088 | 0.7995 | 0.9044 | 1.031 | 1.041 |
| 5.6 | 0.250 | 0.2989 | 0.3482 | 0.3974 | 0.44551 | 0.5363 | 0.68739 | 0.7972 | 0.9037 | 1.033 | 1.042 |
| 5.8 | 0.248 | 0.2966 | 0.3456 | 0.3944 | 0.44231 | 0.5328 | 0.68429 | 0.7951 | 0.9033 | 1.035 | 1.044 |
| 6 | 0.247 | 0.2947 | 0.3433 | 0.3919 | 0.4395 | 0.5297 | 0.68154 | 0.7933 | 0.903 | 1.0371 | 1.045 |
| 6.2 | 0.245 | 0.293 | 0.3413 | 0.3896 | 0.43705 | 0.5271 | 0.67914 | 0.7918 | 0.9028 | 1.0393 | 1.047 |
| 6.4 | 0.244 | 0.2916 | 0.3396 | 0.3877 | 0.43494 | 0.5247 | 0.67706 | 0.7905 | 0.9029 | 1.0415 | 1.049 |
| 6.6 | 0.243 | 0.2904 | 0.3382 | 0.3861 | 0.43314 | 0.5227 | 0.67528 | 0.7894 | 0.9031 | 1.0438 | 1.05 |
| 6.8 | 0.243 | 0.2894 | 0.3371 | 0.3847 | 0.43163 | 0.5211 | 0.6738 | 0.7886 | 0.9034 | 1.0461 | 1.052 |
| 7 | 0.242 | 0.2886 | 0.3361 | 0.3836 | 0.43039 | 0.5197 | 0.67259 | 0.788 | 0.9039 | 1.0485 | 1.053 |
| 7.2 | 0.242 | 0.2881 | 0.3354 | 0.3827 | 0.4294 | 0.5185 | 0.67164 | 0.7877 | 0.9046 | 1.0509 | 1.055 |
| 7.4 | 0.241 | 0.2877 | 0.3349 | 0.3821 | 0.42865 | 0.5177 | 0.67094 | 0.7875 | 0.9054 | 1.0534 | 1.056 |
| 7.6 | 0.241 | 0.2875 | 0.3345 | 0.3816 | 0.42812 | 0.5171 | 0.67047 | 0.7875 | 0.9063 | 1.056 | 1.058 |
| 7.8 | 0.241 | 0.2874 | 0.3344 | 0.3814 | 0.42781 | 0.5167 | 0.67024 | 0.7878 | 0.9074 | 1.0586 | 1.06 |
| 8 | 0.241 | 0.2875 | 0.3344 | 0.3813 | 0.42769 | 0.5165 | 0.67022 | 0.7882 | 0.9086 | 1.0612 | 1.061 |
| 8.2 | 0.242 | 0.2877 | 0.3346 | 0.3815 | 0.42776 | 0.5165 | 0.6704 | 0.7888 | 0.91 | 1.0639 | 1.063 |
| 8.4 | 0.242 | 0.2881 | 0.3349 | 0.3817 | 0.42801 | 0.5167 | 0.67079 | 0.7896 | 0.9115 | 1.0667 | 1.064 |
| 8.6 | 0.243 | 0.2886 | 0.3353 | 0.3822 | 0.42843 | 0.5171 | 0.67137 | 0.7905 | 0.9131 | 1.0695 | 1.066 |
| 8.8 | 0.243 | 0.2892 | 0.336 | 0.3828 | 0.42902 | 0.5177 | 0.67213 | 0.7917 | 0.9149 | 1.0724 | 1.068 |
| 9 | 0.244 | 0.2899 | 0.3367 | 0.3835 | 0.42976 | 0.5185 | 0.67307 | 0.7929 | 0.9168 | 1.0753 | 1.069 |

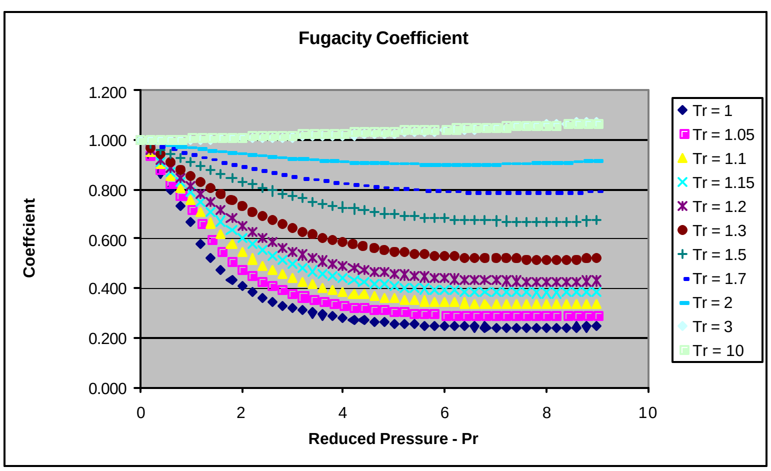


**Figure 6  Comparing Rkfunc to Charts**

| | | |
|---|---|---|
| Tc | 647.4 | Kelvins |
| Pc | 22119247.5 | Pascals |

| | Tr | Pr | Kode | Values |
|---|---|---|---|---|
| Compressibility | 1.4 | 1.51 | 1 | 0.824636 |
| Enthalpy Departure | 1.23 | 1.37 | 2 | 9.648814 |
| Entropy Departure | 1.23 | 1.37 | 3 | 5.661568 |
| Fugacity | 1.7 | 1.49 | 4 | 0.916666 |

# Web-Based Instructional Tools for Heat and Mass Transfer

**Haishan Zheng and Jason M. Keith**
**Department of Chemical Engineering**
**Michigan Technological University**
**Houghton, MI 49931**

Abstract

This paper demonstrates four web-based instructional tools for heat and mass transfer based on JAVA applets. These tools are closely related to topics in two chemical engineering courses: transport phenomena and chemical reactor design. They simulate four classic problems which are one dimensional unsteady mass diffusion, one dimensional unsteady-state heat conduction in slabs, cylinders and spheres, Heisler charts for unsteady state heat transfer, and reaction and diffusion in porous catalytic substrates. They provide students with a dynamic observation of heat and mass transfer process unavailable in traditional textbooks. Educators and instructors can use these tools to enhance students understanding of the concepts of heat and mass transfer.

Introduction

As many efforts are being applied to the introduction of the World Wide Web (WWW) in the chemical engineering curriculum, WWW shows a growing significance in the engineering classroom, especially with the advent of web-based instructional tools. These web-based instructional tools are JAVA applet programs that are run by the JAVA Virtual Machine that is integrated in the web browser. However, web-based instructional tools for fundamental chemical engineering courses such as heat and mass transport are seldom found online.

Since the Java language and Virtual Machine have sufficient mathematical capability for mathematical simulations of mass and heat transfer, it is reasonable to develop web-based instructional tools to simulate classic problems from these fields. With the aid of these web-based instructional tools, students can capture the basic principles of these transport processes by observing dynamic phenomena that are often difficult to explain in textbooks. Observing these processes gives students valuable insight into heat and mass transfer concepts. Furthermore, the interactive nature of these web-based instructional tools may improve the motivation of students, which could be a significant factor in achieving student success.

This paper will describe four JAVA applets developed for simulation of basic heat and mass transfer processes. These applets are a part of “Web Instructional Tools for Engineering,” a one year project funded by the Michigan Space Grant Consortium. These four JAVA applets were to be developed with two goals in mind: to be visual enough to enable students to capture transport concepts, and to be interactive to keep the students’ interest.

Description of JAVA applets

1. One dimensional unsteady mass diffusion

*Introduction*

Mass transfer is a crucial issue in the transport and unit operations course. Students are expected to learn the concept that mass transfer is due to a difference in concentration between two points. They also need to understand what a diffusion coefficient is and how it affects the process of mass diffusion.

In order to achieve these goals, a simple mass diffusion experiment, a solute diffusing through a stagnant fluid, is simulated in this applet. Diffusion of solutes in a stagnant fluid occurs in many industrial processes. One unique application of this mechanism is the manufacture of single crystal semiconductor materials with the design objective of optimizing the distribution of dopant molecules in a crystal. Often, fundamental experiments are performed in the microgravity environment of space [1]. Therefore, it is a very good example to demonstrate the mass diffusion concept.

*Model description*

As figure 1 shows, the concentration is initially zero at $x<L/2$, and is equal to $C_0$ at $x>=L/2$.

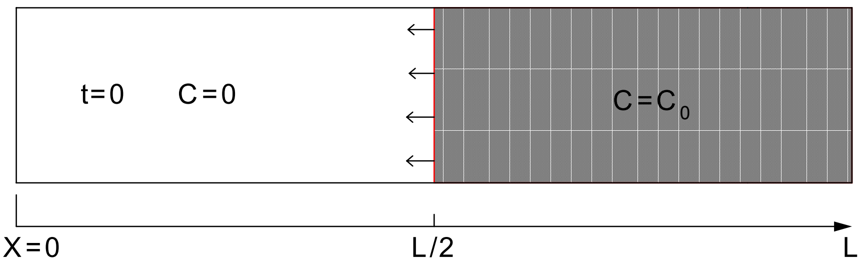



**Figure 1. Schematic of a microgravity semiconductor dopant diffusion experiment**

With increasing time, the solute will diffuse from the right side to the left side of the tube. This unsteady mass diffusion process can be described by the following one dimensional mathematical model [1],

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} \tag{1}$$

The initial and boundary conditions are

$$\text{at} \quad t = 0, \qquad x < L/2, \qquad C = 0 \tag{2}$$

$$\text{at} \quad t = 0, \qquad x >= L/2, \qquad C = C_0 \tag{3}$$

$$\text{at} \quad t = t, \qquad x = 0 \ \text{ and } \ x = L, \qquad \frac{\partial C}{\partial x} = 0 \tag{4}$$

*Graphical User Interface (GUI) of the applet*

As mentioned above, a web-based instructional tool should keep the student's interest and allow for them to visualize the mass transfer process. To build an effective web-based instructional tool, the GUI of the applet must be designed to be user friendly.

As figure 2 shows, the GUI of the applet can be divided into three areas: display, parameter input, and control. The display area is the main area of the interface where the simulation of the mass diffusion process and the concentration profile are displayed. Therefore, this area occupies the most space.

Parameter input is at the right side of the display area. Several key parameters for mass diffusion such as diffusion coefficient and tube length are listed here. The control area is at the bottom of the interface which is separated from the other two areas by a yellow bar. In order to keep the control unit as simple as possible, there are only three buttons in the control area: start, stop, and pause/resume. These three sections are organized in such a way as a game-like environment. With such a design, students can easily run the applet, even without reading instructions.

*Functions and features of the applet*

This applet leads the student through the exploration of short-time, unbounded diffusion and long-time diffusive processes with a no-flux condition. The student inputs a set of parameters and presses the "Start" button. The button activates a graphic simulation of mass diffusion in the tube. The clock at the top of the display area displays the simulated time elapsed. Under the clock, a contour plot and graph of the concentration profile are shown.

This dynamic animation illustrates the nature of the mass diffusion process in the tube. Watching this animation is just like watching a real mass diffusion experiment. This can greatly aid student understanding of the basic principles of mass diffusion.

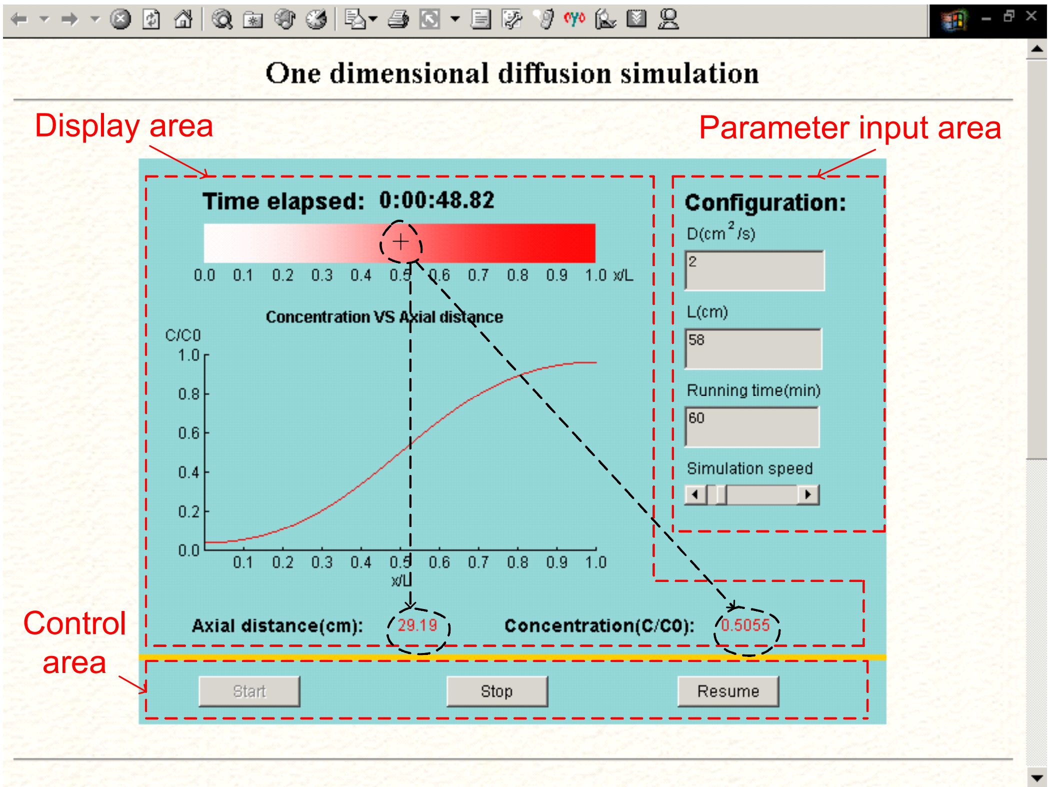



**Figure 2. Screenshot of the JAVA applet for one dimensional mass diffusion**

To visualize the mass diffusion process, the color in the tube represents the concentration of solute at that position. In the JAVA language, a specific color can be expressed by three integer values for red, green and blue (R, G, B) which range between 0 and 255. In this applet, red (R=255, G=0, B=0) represents the initial concentration $C_{0,}$ and white (R=255, G=255, B=255) represents zero. The color for any concentration C between $C_0$ and 0 is calculated by the following equations.

$$R = 255 \quad \text{and} \quad G = B = 255\left(1 - \frac{C}{C_o}\right) \tag{5}$$

One key feature is that this applet allows the student to specify the diffusion coefficient and the tube length. This interactive feature not only keeps the student's interest when he/she is running this applet, but also allows the student to find out how these parameters affect the mass diffusion process.

Another useful feature is that this applet allows the mass diffusion process to pause at any time. As figure 2 shows, the student can obtain the x and concentration values of every position in the tube simply by moving the cursor in the color bar during the pause status.

Another welcome feature is that this applet allows the student to adjust the speed of simulation. This can significantly reduce the time for completing simulations in such instances as the student wants to run a long time simulation to estimate the time to reach steady-state or run a simulation in which the solute diffuses in a fluid with a very small diffusion coefficient.

2. One dimensional unsteady-state heat conduction in slabs, cylinders and spheres

*Introduction*
Heat conduction is another key concept in transport phenomena because of the larger number of heating and cooling problems occurring in industrial processes. Unsteady state heat conduction in various geometries is the classic example for demonstration of the heat conduction concept. However, students must refer to cumbersome charts to estimate local temperatures without visualizing the process. The second JAVA applet will dynamically simulate the heat conduction in slabs, cylinders and spheres,

*Model description for unsteady-state conduction in a large flat plate [2]*

As figure 3 shows, unsteady state heat conduction occurs in a large flat plate of thickness 2L in the x direction and having infinite dimensions in the y and z direction. The plate with original uniform temperature $T_0$ is exposed to an environment at temperature $T_1$ at time t=0. Heat conduction occurs only in the x direction.

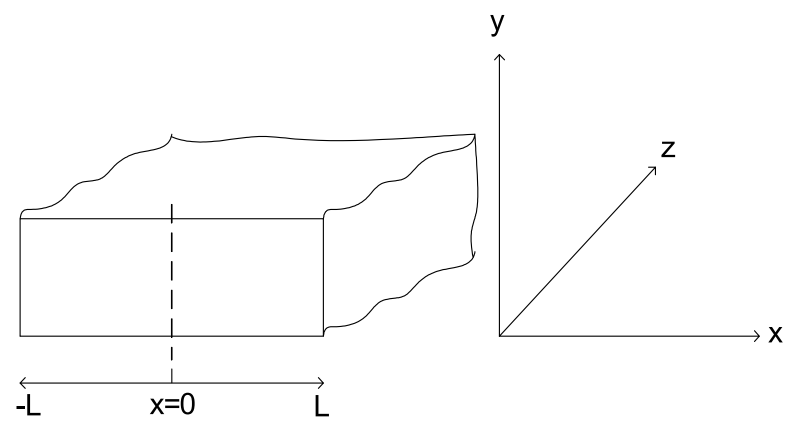



**Figure 3. Schematic of unsteady-state conduction in a large flat plate**

The mathematical model to describe this heat conduction problem is

$$\rho C_P \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \tag{6}$$

with boundary and initial conditions that:

$$\text{at} \quad t = 0, \qquad T = T_0 \tag{7}$$

$$\text{at} \quad x = 0, \qquad \frac{\partial T}{\partial x} = 0 \tag{8}$$

$$\text{at} \quad x = \pm L, \qquad h(T - T_1) = -k\frac{\partial T}{\partial x} \tag{9}$$

Similar models are developed for a long cylinder and for a sphere [2].

*Graphical User Interface (GUI) of the applet*

As figure 4 shows, this applet has a similar GUI design as the mass transfer applet. Everything is put onto one screen to make this applet as easy as possible to understand and run. Although this applet simulates heat conduction in three geometries, the same GUI is used for three geometries except some changes of the labels.

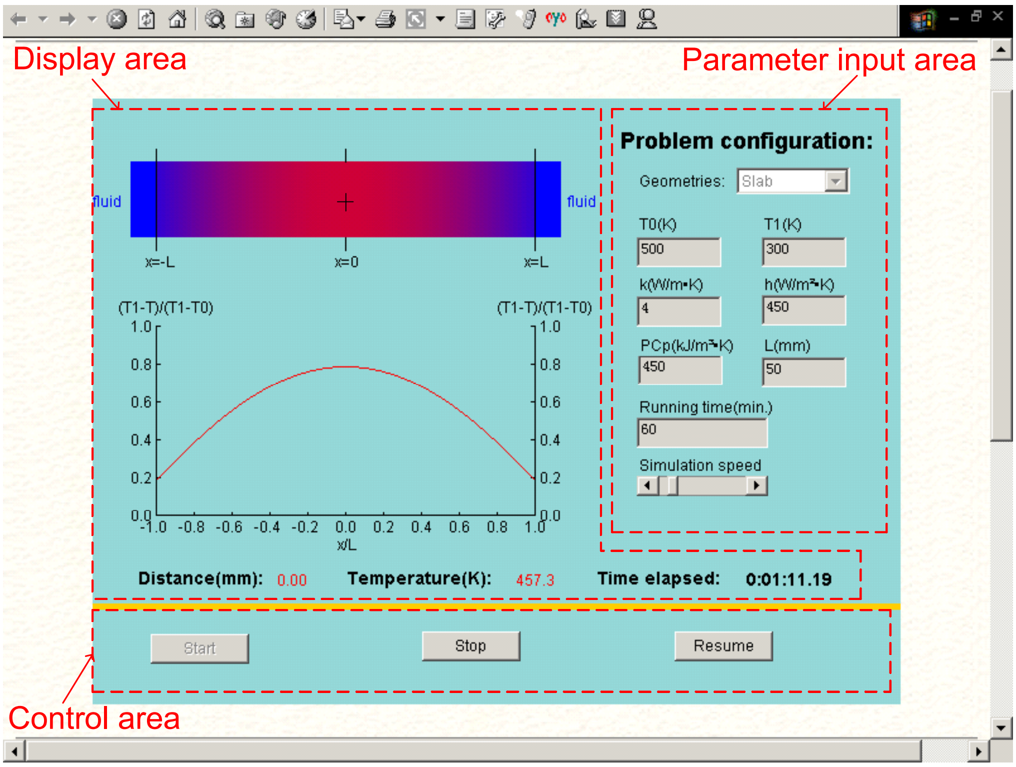



**Figure 4. Screenshot of the heat conduction JAVA applet**

*Functions and features of the applet*

This applet is a collection of three similar heat conduction models: heat conduction in slabs, cylinders, and spheres. The simulations for these models together help students grasp the principles of heat conduction.

The student first chooses the geometry where heat conduction occurs. The geometry is set as a slab by default. As figure 5 shows, a colored bar represents the cross section of slab, and a colored circle represents the cross section of a cylinder or a sphere. The student enters a set of initial conditions and parameters and presses the “Start” button to initiate the simulation. Later, the student can press the “Pause” button to pause the simulation or the “Stop” button to stop the simulation.

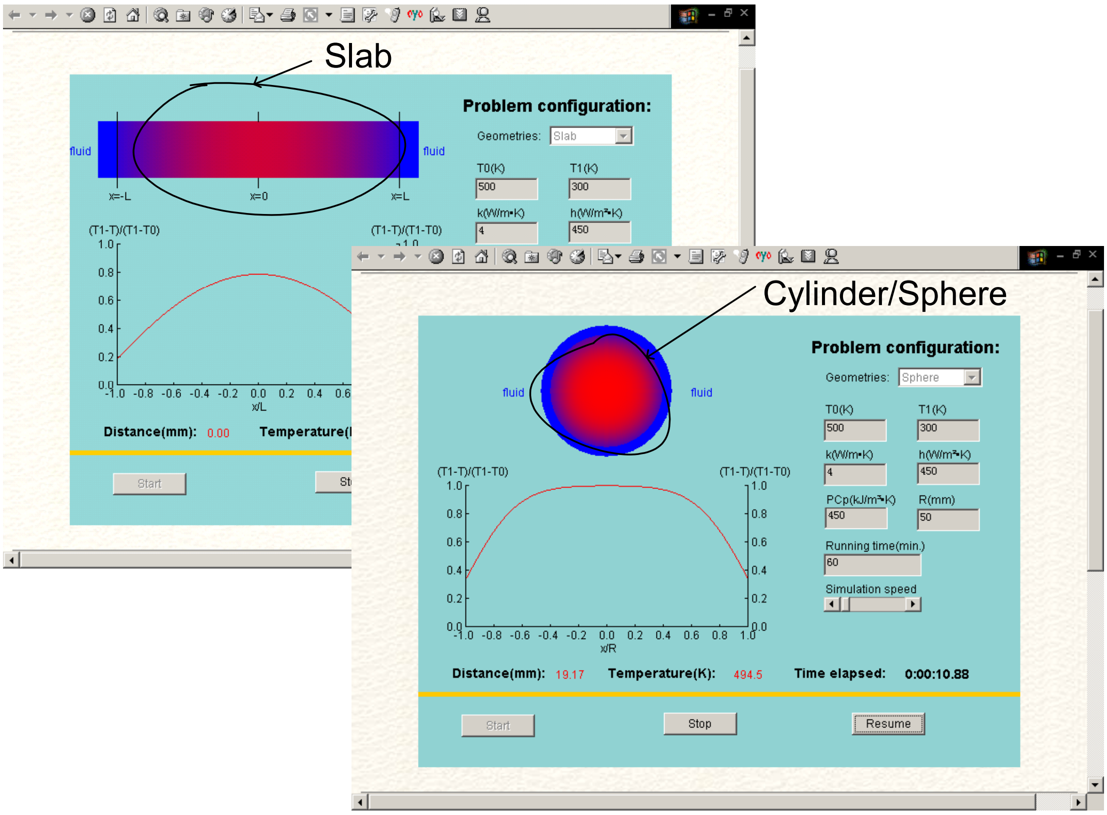



**Figure 5. Screenshot of the heat conduction JAVA applet**

The different colors represent different temperatures. $T_0$ and $T_1$ are respectively the initial solid temperature and the fluid temperature. To keep consistency, red always represents the higher temperature between $T_0$ and $T_1$, and blue (R=0, G=0, B=255) always represents the lower temperature. Therefore, there are two instances for the color representing any temperature T between $T_0$ and $T_1$.

$$\text{if } T_0 > T_1, \quad R = 255 \times \frac{T_1 - T}{T_1 - T_0}, \quad G = 0 \ \text{ and } \ B = 255 \times \left(1 - \frac{T_1 - T}{T_1 - T_0}\right) \tag{10}$$

$$\text{if } T_0 < T_1, \quad R = 255 \times \left(1 - \frac{T_1 - T}{T_1 - T_0}\right), \quad G = 0 \ \text{ and } \ B = 255 \times \frac{T_1 - T}{T_1 - T_0} \tag{11}$$

In addition to visualizing the heat conduction process using color, a temperature profile is shown simultaneously during the simulation to strengthen the visual effect of the simulation of the heat conduction process.

The input parameter interface allows the student to input different heat transfer coefficients, thermal conductivities and heat capacities. The student can learn how these parameters affect the heat conduction through the comparison of the different simulation results. For example, the student can verify when internal resistance is negligible ($Bi=hL/k < 0.1$) with this applet.

Another feature mentioned in the previous applet is that this applet also allows the student to suspend the simulation and move the cursor in the color bar or circle to find out the temperature at any position in the solid at the elapsed time.

3. Heisler charts for unsteady state heat transfer [2]

*Introduction*

Analytical solutions are available for the above three heat conduction models. Therefore a chart for determining the temperatures at any position in the solid and at any time can be constructed for a convenient reference to practical applications. Although such charts can be found in textbooks, an online implementation is easier to use due to the unprecedented popularity of WWW and the power of JAVA to determine the exact coordinates on the graph. This applet implements the charts for slabs, cylinders, and spheres.

*Graphical User Interface (GUI) of the applet*

As figure 6 shows, the GUI of this applet is a TV-like design with a big screen at the left side and a small control panel at the right side. The screen occupies almost 70% of the interface area. On the control panel, there is a list of geometry choices, which the student can use to choose the channel he/she wants to look: slab, cylinder, or sphere. There are also such control buttons on the control panel. Two are for adjustment of the key parameters m (a ratio of conduction to convection) and n (a dimensionless distance). The other four buttons adjust the position of graph.

*Functions and features of the applet*

This online chart is very easy to use. Unlike the charts in the textbook that show many lines simultaneously, this chart shows only one line at one time. This gives the student a clear look and allows the student to determine the value quickly.

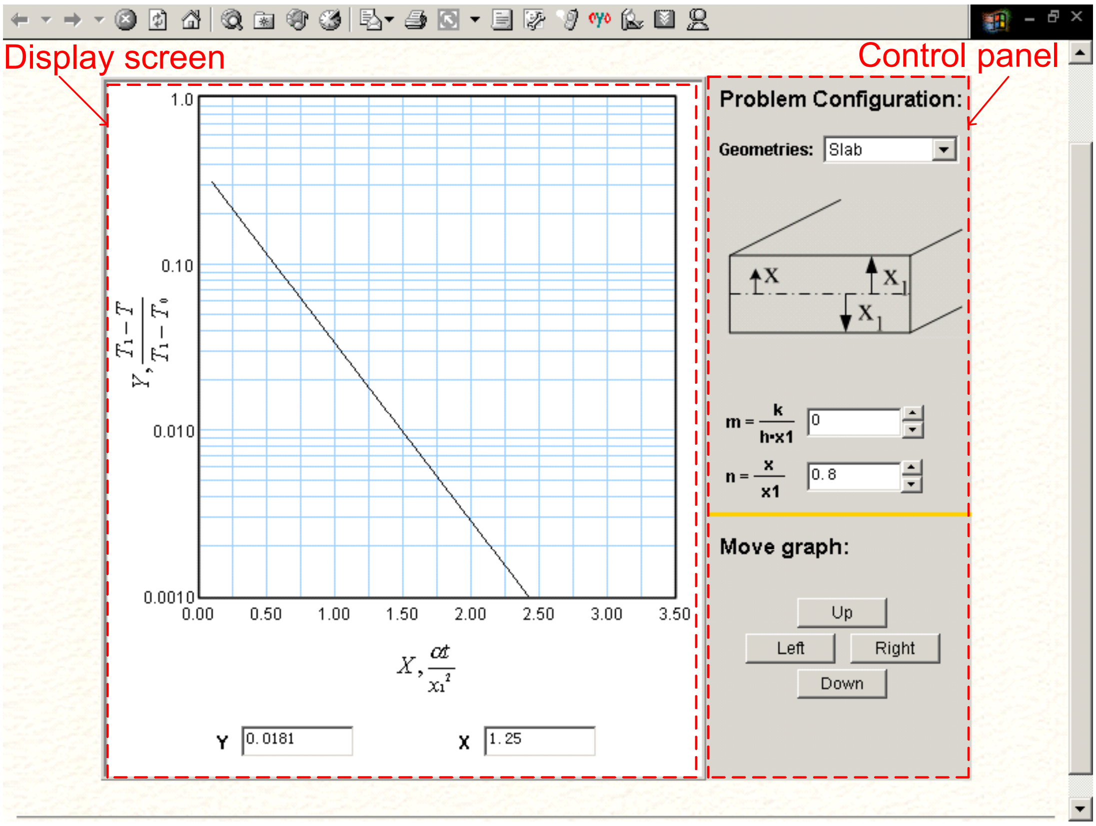



**Figure 6. Screenshot of JAVA applet for the Heisler charts**

Originally, the geometry is set as a slab by default. The parameters m and n are set as 0.0 and 0.8, respectively. The student can use the control panel to make his/her own choice. For the parameters m and n, the student can input directly the desired value in the text field or increase/decrease the value by pressing the buttons.

This applet allows the student to move the graph by press the “Up”, “Down”, “Left”, and “Right” buttons. As figure 7 shows, double clicking the chart calls an axes properties dialog box. This allows the student to zoom in and out the graph.

One nice feature is that the student can obtain the values of the dimensionless time, X and the dimensionless time, Y from the “X” and “Y” text field simply by moving the cursor to the desired position on the line (shown in figure 7). This applet also can be used as a calculator. A relative value of X can be obtained by input of a value in the “Y” text field and pressing “Enter” and vice versa. This gives the student a quicker and more precise way to determine the dimensionless temperature or concentration than by using the charts in the textbook.

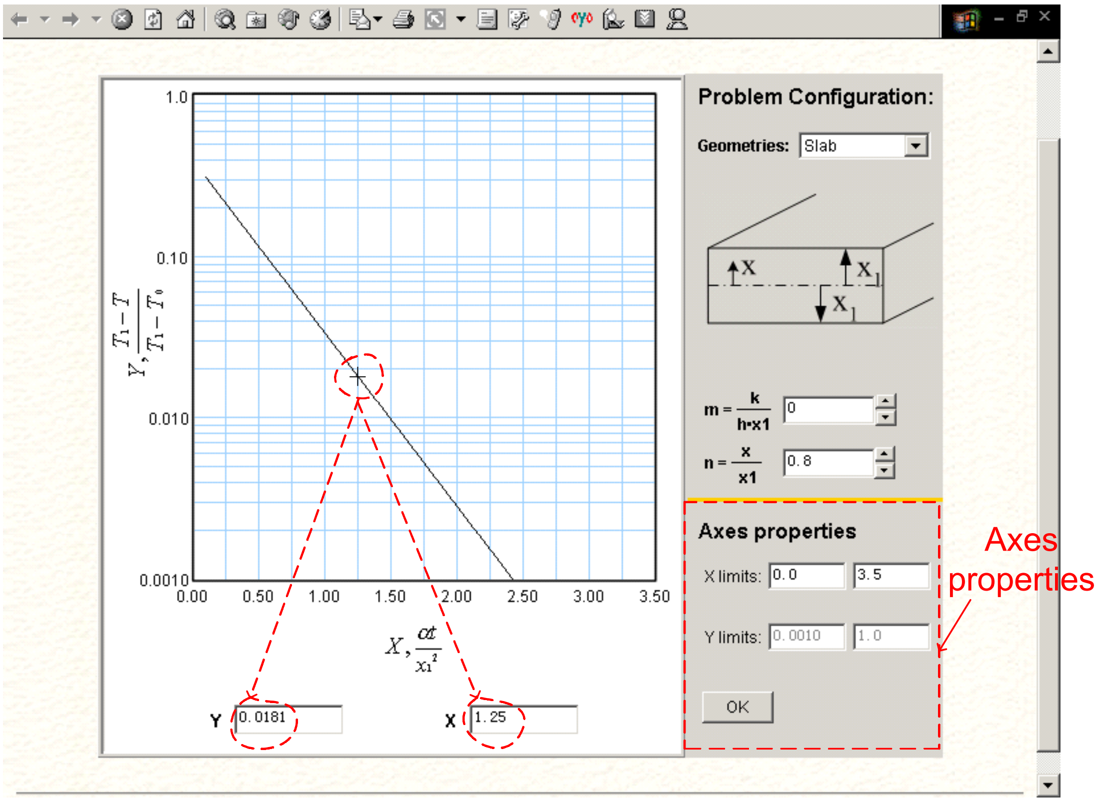



**Figure 7. Screenshot of the JAVA applet for Heisler charts**

4. Reaction and diffusion in porous catalytic substrates [3]

*Introduction*

Reaction and diffusion in porous catalytic substrates are demonstrated in transport and reactor design courses. Although analytical solution for catalysts with simple geometries such as slabs, cylinders and spheres are shown in textbooks, a graphical solution is more suitable to illustrate the reaction and the diffusion in catalysts. This is achieved by this JAVA applet.

*Graphical User Interface (GUI) of the applet*

As figure 8 shows, this applet has the same TV-like GUI as the previous applet. The difference is that there are buttons to choose the concentration profile within the catalyst pellet or the effectiveness factor plot as a function of the Thiele modulus.

*Functions and features of the applet*

This applet can show the concentration profile and the effectiveness factor plot. The student chooses the geometry and enters the values of the following parameters: the diffusion coefficient D, the first order reaction rate constant k and the characteristic length R of the catalyst pellet. Pressing the “Concentration profile” button illustrates the concentration profile in the catalyst pellet.

On the concentration profile screen, the student can obtain the concentration at every position in the catalyst pellet by moving the cursor on the concentration profile (shown in figure 8). The student also can directly get the concentration value simply by input of the position value in the “r” text field and pressing “Enter” and vice versa.

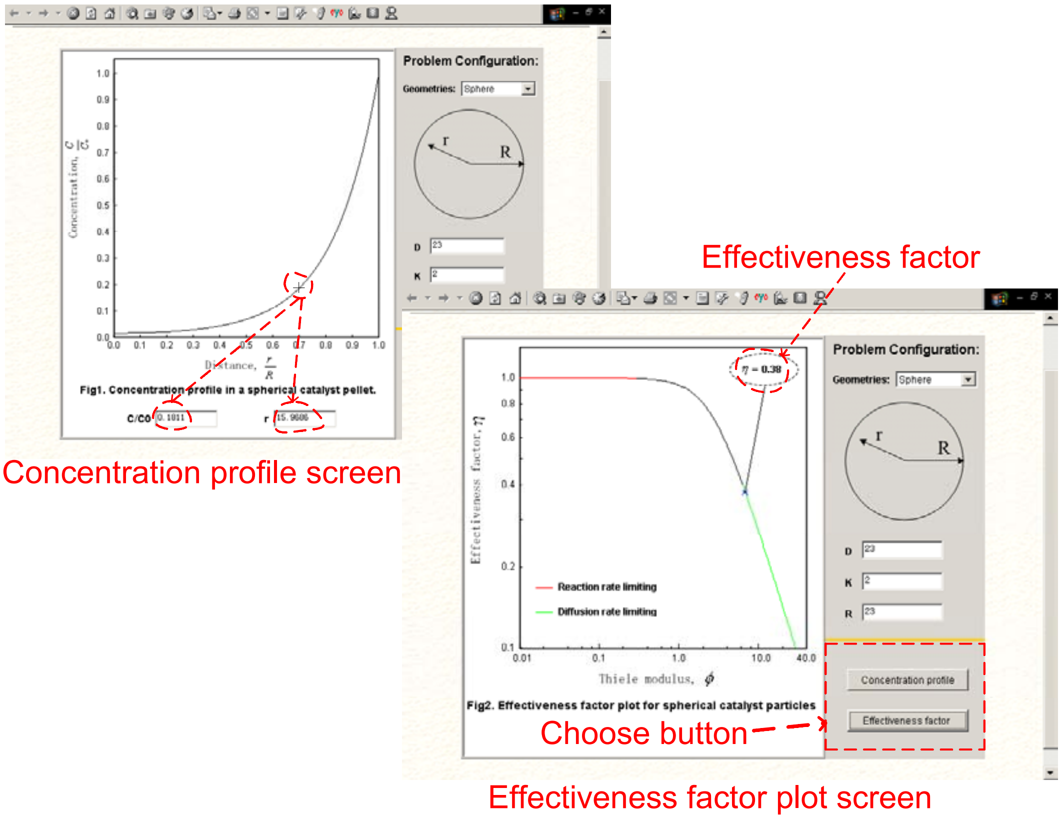



**Figure 8. Screenshot of the JAVA applet for diffusion and reaction in porous catalysts**

When the student presses the “Effectiveness factor” button, the screen changes to the effectiveness factor plot. On this screen, the applet shows the effectiveness factor plot and points out the value of the effectiveness factor for this specific instance (shown in figure 8). The effectiveness factor plot uses a red color to indicate the reaction rate limiting area and a green color to indicate the diffusion rate limiting area. Therefore, the student can learn to which limiting area this instance belongs, and understand how diffusion and reaction rate affect the reaction within porous catalysts.

Impact on learning

The web based tools create possibilities for students to learn effectively. Felder [4] gives a good review of four learning style models (Myers-Briggs Type Indicator, Kolb’s Learning Style Model, Hermann Brain Dominance Instrument, and the Felder-Silverman Learning Style Model). Kolb’s Learning Style Model says that learning includes four processes, which are: concrete experience, reflective observation, abstract conceptualization, and active experimentation [5]. The ideal teaching would include all the four processes. For example, to illustrate the concept of heat conduction, the instructor begins with a heat conduction experiment. Next, the instructor reflects on this experiment, explaining the meaning, and then applies the meaning into abstract mathematical equations. Finally, the instructor gives students homework with similar problems. Additional information about the Kolb model is available in the literature [6-8].

Unfortunately, the instructor usually cannot perform suitable heat conduction experiments in the classroom. Showing data may be useful but not as effective as a computer-based applet that the students can relate to. The web based tools developed here permit the instructor to show real heat transfer processes in the classroom. This will greatly enhance two of Kolb's learning processes: concrete experience and reflective observation. Furthermore, students can explore more problems through the simulation of the web based tools after class. This allows students to explore the concept of heat conduction through active experimentation. Therefore, the web based tools improve learning by provide the possibility for students to make use of all the four learning processes.

The web-based tools are being used by J. Keith as auxiliary teaching tools in CM3120: Transport / Unit Operations 2, during the Spring 2003 semester at Michigan Technological University. This course focuses on the fundamentals of heat and mass transfer. Several demonstrations and homework assignments, using these web-based tools, will be applied during the course. To illustrate the potential use of these tools, an example that has been already used will now be described.

Example Problem: Cooking a Slab of Meat

In this exercise, students were asked to solve the following problem "cooking a slab of meat" using the Heisler charts in the textbook, the web-based java applet "One dimensional unsteady-state heat conduction in slabs, cylinders and spheres", and the web-based java applet "Heisler charts for unsteady state heat transfer". The problem is modified from number 5.3-5 from the 3rd Edition of Geankoplis' textbook [2].

> A slab of meat 25.4 mm thick originally at a uniform temperature of 10 $^oC$ is to be cooked from both sides until the center reaches 121 $^oC$ in an oven at 177 $^oC$. The convection coefficient can be assumed constant at 25.6 $W/m^2K$. Neglect any latent heat changes. The thermal conductivity is 0.69 W/mK and the thermal diffusivity is $5.85 \times 10^{-4}$ $m^2/h$.
>
> a) Calculate the time $t_{cook}$ required to reach 121 $^oC$
> b) Calculate the surface temperature of the meat at time $t_{cook}$

The results obtained from the two web-based java applets are very close to the results obtained from the charts in the textbook. Both java applets provide the student with a quicker and easier way to solve the problem than using the charts in the textbook. The java applet "One dimensional unsteady-state heat conduction in slabs, cylinders and spheres" also offers the visualization of the heat conduction in meat. This characteristic allows the student to understand the diffusive motion of the heat in the media. Both the java applets allow the student to change the convection coefficient, thermal conductivities and heat capacities, and quickly see what effect these changes have on the system. The student benefits from this characteristic to quickly explore the relationships between these parameters and the heat conduction process without having to do lengthy calculations. Therefore, the two java applets enhance the student understanding the concept of the heat conduction.

To measure the effectiveness of using the java applets, students are also required on their homework to write a statement about the utility of the java applets. The students seemed to genuinely enjoy using the java applets. When using the java applets, students felt more motivated to simulate some different problems by changing the parameters. Student also found the java applets to be very useful in understanding the concept of the heat conduction. Shown below are quotes from the students:

- "The conduction web tool is neat to watch. You can actually see how different convection coefficients, thermal conductivities, and thermal diffusivities affect the temperature at the different points in your object. It is easier to learn these things by visualizing the effects and not just starting at the equations. They are excellent teaching aids".

- "The animation gives the user a good idea of how the heat conduction takes place, changing from blue to red as the object heats up. These tools are very easy to use and provide the added element of visualization. You can't get that in a book."

- "If this were my applet that I designed I would get it patented immediately for it is useful. Hopefully it will remain on the web while I am at my co-op so I can demonstrate it's simplicity and accuracy with minimal time."

## Conclusions

This paper demonstrates four web-based instructional tools for heat and mass transfer that are based on JAVA applets. These web-based instructional tools closely relate to instruction topics in transport phenomena and reactor design courses. They can be accessed for free at the web address: http://www.chem.mtu.edu/~jmkeith/webtools. Educators and instructors can use these web-based tools in the classroom to instruct students and enhance their understanding. They were first used in the spring semester of 2003 at Michigan Technological University, and student feedback has been extremely positive.

## Acknowledgements

The authors want to acknowledge the financial support provided by the Michigan Space Grant Consortium.

## Bibliography

1. W. A. Arnold, D. H. Matthiesen, and J. M. Keith, "Numerical Simulation of Soret Diffusion Effects Using a Shear Cell," AIAA Paper 95-0502.
2. C. J. Geankoplis, *Transport Processes and Unit Operations, Third Edition*, Englewood Cliffs, NJ: Prentice Hall (1993).
3. H. Scott Fogler, *Elements of Chemical Reaction Engineering, Second Edition*, Englewood Cliffs, NJ: Prentice Hall (1992).
4. R. M. Felder, "Matters of Style," *ASEE Prism*, **6(4)**, 18 (1996).
5. D. A. Kolb, *Experiential learning: Experience as the source of learning and development*, Englewood Cliffs, NJ: Prentice Hall (1984).
6. B. McCarthy, "The 4MAT System: Teaching to Learning Styles with Right/Left Mode Techniques," Barrington, IL: EXCEL, Inc. (1987).
7. J. E. Stice, "Using Kolb's Learning Cycle to Improve Student Learning," *Engr. Education*, **77**, 291 (1987).
8. J. N. Harb, S. O. Durrant, and R. E. Terry, "Use of the Kolb Learning Cycle an the 4MAT System in Engineering Education," *J. Engr. Education*, **82(2)**, 70 (1993).

HAISHAN ZHENG
Haishan Zheng is a PhD candidate in chemical engineering at Michigan Technological University. He received his BS in 1993 and MS in 1996 from Beijing University of Technology, China. Before studying at Michigan Technological University, he served as chemical engineer at Guangdong Petrol-Chemical Research Institute, China.

JASON M. KEITH
Jason Keith is an Assistant Professor of Chemical Engineering at Michigan Technological University. He received his PhD from the University of Notre Dame in August 2000. Jason teaches a new senior elective design course, Interdisciplinary Design, and a required junior level Transport / Unit Operations II course. Jason has been married for 2 years and his puppy dog "Pennant" is now one year old.

**GAMS Newsletter Number 12**
**Bruce McCarl**

## New Release - GAMS 21.0

GAMS Corporation has just released version 21.0 of GAMS. The main features visible to the user (a lot of work was done on internal handling of functions) are several bug fixes, a few language enhancements, documentation enhancements, IDE enhancements, GDX enhancements, solver upgrades and some new solvers.

Bug fixes involve repairs in a couple of GAMS features plus GAMSBAS, GAMSCHK, GDXXRW, XLIMPORT, XLEXPORT and XLDUMP. If you have had unresolved problems with those in the past you might want to try them again. The release notes elaborate at http://www.gams.com/docs/release/release.htm.

Language enhancements involve the ability to specify a multidimensional set via a table as discussed in the Sets chapter of the McCarl documentation and uses syntax like set Table Linked by road 3 (origins, destinations) places linked by roadways Portland London Houston.

Documentation enhancements involve the update of the solver guides, the integrated system release of the McCarl User Guide as discussed in the last newsletter (http://www.gams.com/mccarl/newsletter/news11.htm) and new documentation access features through the Help menu in the IDE.

IDE enhancements involve the afore mentioned help path to documentation, inclusion of a GDX differencing utility; development of GDX viewing capability as discussed in the McCarl User Guide chapter on Using GAMS Data Exchange or GDX Files under the Identifying GDX file contents section; and improved memory of window sizes.

GDX enhancements involve the afore mentioned IDE viewer plus the capability to pass selected attributes of variables and equations like solution levels or marginals.

Solver upgrades: A new version of CPLEX (8.1) is now available that has new QP, MIP, MIQP and Parallel computer LP solving capabilities as discussed in http://www.gams.com/docs/release/rel_cplex.htm. CONOPT3 is now the default version with its improved scaling features among other items. OSL3 is now the default version with its improved memory management, simplex, and barrier solvers. MINOS, SNOPT, PATH, and XPRESS have all been updated - see the release notes at http://www.gams.com/docs/release/release.htm for details.

New Solvers are available for purchase including: Global optimizers for general NLP and MINLP problems including BARON, LGO and OQNLP. MOSEK is now available, as is NLPEC, the first solver shipped for MPEC problems (Math Programs with Equilibrium Constraints). NLPEC automatically reformulates the MPEC model as an NLP, solves the NLP, and recovers the MPEC solution. All of these but NLPEC are subjects of solver manuals accessible through the IDE help.

## Did you know:

The new documentation explains a lot of things not previously covered so here I point a few out.

Through use of GDX utilities you can use an EXCEL spreadsheet to create graphics images. You can also use spreadsheet capabilities to do numerical procedures not in GAMS like estimation of a regression. All are illustrated in the documentation chapter on Links to Other Programs Including Spreadsheets in the sections on Spreadsheet graphics and Interactively including results.

**Courses offered**

I will teach Advanced GAMS in Hamburg Germany, June 10-13, 2003 and in Texas January 12-15, 2004. A basic course will be scheduled for May, 2004. Further information and other courses are listed on http://www.gams.com/courses.htm.

May 29, 2003

# GAMS Newsletter Number 13
**Bruce McCarl**

## New Release - GAMS 21.2

GAMS Corporation is just in the process of releasing version 21.2 of GAMS. It is mainly a maintenance release but does include a beta version of a replacement for the GAMSBAS basis saving program. More details will follow soon.

## Ranking Program

Tom Rutherford and Paul van der Eijk just released a procedure for sorting arrays in GAMS. Details are available through the web page http://debreu.colorado.edu/gdxrank/.

## Did you know:

A tuple (multi dimensional set) can be used as follows:

```
set link(i,j) linked areas;
link(i,j)$(supply(i) and demand(j) and distance(i,j))=yes;
shiplim(link(i,j)).. ship(i,j)=l=10;
```

where the tuple defines the i,j cases that will have a constraint defined for them.

## Global Optimization Workshop

A workshop is being held in Washington on September 18 on Global Optimization. The workshop will introduce the recently released GAMS global solvers - BARON, LGO, OQNLP - and the MProbe model analyzer. Within this context, the general subject of global optimization and a wide range of its current and potential applications will also be reviewed. For more information see http://www.gams.com/courses/goworkshop.htm.

## Courses offered

I will teach Basic GAMS in Colorado Springs October 6-9, 2003 and Advanced GAMS in Texas January 12-15, 2004. Further information and other courses are listed on http://www.gams.com/courses.htm.

This newsletter is not a product of GAMS Corporation although it is distributed with their cooperation.

September 8, 2003

# FOCAPD 2004

Foundations of Computer Aided Process Design

## July 11-16, 2004

The Friend Center
Princeton University, New Jersey

http://www.focapd.org/FOCAPD2004

The major theme of FOCAPD 2004, Discovery Through Product and Process Design, reflects this remarkable shift in the industrial sector. Princeton University Professor Christodoulos A. Floudas and Dr. Rakesh Agrawal of Air Products and Chemicals, chair this conference with a goal to create an academic and industrial dialogue, a critical assessment of existing enabling technologies, a discussion on research, education, and industrial needs, and a forum of new directions, challenges and opportunities in product and process design.

FOCAPD 2004 will be held at the Friend Center at Princeton University, Princeton, New Jersey, July 11-16, 2004. This international conference will attract world-renowned experts from academia and industry, researchers and practitioners from government laboratories, product and processing industries, technology and consulting companies, and graduate students. Please check out the above website for more information.

# CACHE
# STANDARD ORDER FORM

| CACHE Product Description | Qty. | Supporting | Non-Supporting | Total |
|---|---|---|---|---|
| AI Monograph - Volumes 1, 2, 3 & 4 (sold separately) | | $20 | $25 | |
| AI Monograph Set (Volumes 1-4) | | $50 | $75 | |
| CD-ROM - Volume 2<br>Student chapters and supporting departments<br>Individual students and faculty<br>Individuals with no academic connection | | <br>$15 | <br>$20<br>$20<br>$50 | |
| CD-ROM - Volume 3<br>Student chapters and supporting departments<br>Individual students and faculty<br>Individuals with no academic connection | | <br>$15 | <br>$20<br>$20<br>$50 | |
| Chemical Engineering Problems with Solutions – CD ROM | | $10 | $10 | |
| ChemSep Version 3.71<br>• without documentation<br>• with documentation<br>**(License agreement must be signed first)** | | <br>$100 + annual $60<br>$135 + annual $60 | <br>$115 + annual $75<br>$150 + annual $75 | |
| Computers in Chemical Engineering Education | | $15 | $15 | |
| EMR Technology Library | | $25 | $30 | |
| KORF - Fluid Flow via Graphics<br>**(License agreement must be signed first)** | | $50 per yr. | $60 per yr. | |
| Interactive Computer Modules<br>• per course<br>• set of four | | <br>$35<br>$100 | <br>$75<br>$200 | |
| MultiBatch<u>DS</u><br>Two (2) 3.5" PC disks<br>Manual<br>**(License agreement must be signed first)** | | <br>$90 + annual $50<br>$10 | <br>$120 + annual $75<br>$10 | |
| Multimedia-Based Educational Applications of Computer Simulations of Chemical Engineering Processes (Dow, Amoco, Eastman, Mobil) | | $200 per module plus annual $25 | $200 per module plus annual $25 | |
| Multimedia Education Laboratory<br>Chemical Engineering Fundamentals in Biological Systems<br>Material & Energy Balances Multimedia Modules<br>Visual Encyclopedia of ChE Equipment Version 2.1 | | <br>$25<br>$25<br>$25 | <br>$25<br>$25<br>$25 | |
| Open-Ended Problems in Chemical Reaction Engineering | | $10 | $15 | |
| Polymath Version 5.1 | | $125 + annual $75 | $150 + annual $100 | |
| Process Design Case Study - Volume 1<br>Process Design Case Study - Volume 2<br>Process Design Case Study - Volume 3<br>Process Design Case Study - Volume 4<br>Process Design Case Study - Volume 5<br>Process Design Case Study - Volume 6 (with GAMS)<br>Process Design Case Study – Volume 7 (binder & disk)<br>OR (CD-ROM) | | $30<br>$30<br>$30<br>$30<br>$30<br>$110<br>$50<br>$30 | $50<br>$50<br>$50<br>$50<br>$50<br>$150<br>$90<br>$50 | |
| Strategies for Creative Problem Solving<br>• personal use only<br>• domestic universities<br>• overseas universities<br>• industrial companies | | <br>$65<br>$90<br>$90 | <br>$65<br>$90<br>$105<br>$210/machine | |
| TARGET II<br>**(License agreeement must be signed first)** | | $5 | $5 | |
| THEN | | $5 | $5 | |
| | | | | |
| OTHER PRODUCTS (enter name and price if known): | | | | |
| | | | | |
| | | | **TOTAL** | |

**<u>PREPAYMENT IS REQUIRED</u>**

Cost of international shipping is determined
by destination and weight of order.

**<u>TO ORDER</u>:**

Please fill out information below and fax to:
(512) 295-4498

**<u>Payment</u>** (please check )

___ check (US funds, US bank)

___ Purchase Order (domestic only)

___ Credit Card

Mastercard or Visa ____________________________________

Card No: _____________________________________________

Exp Date: ____________________________________________

Exact Name on Card: __________________________________

Name: ________________________________________________________________________________________

Address: ______________________________________________________________________________________

_______________________________________________________________________________________________

_______________________________________________________________________________________________

Phone: ________________________________________ Fax: ____________________________________________

E-mail address: _______________________________________________________________________________

Your signature: ___________________________________________________

**Make checks payable to:**

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

**E-Mail:** cache@uts.cc.utexas.edu **Fax:** (512) 295-4498 **Phone:** (512) 295-2708