CONCEPTUAL DESIGN AND PROCESS SYNTHESIS

James M. Douglas
University of Massachusetts
Amherst, MA 01003

Jeffrey J. Siirola
Eastman Chemical Company
Kingsport, TN 37662

Abstract

Systematic approaches for the invention of conceptual chemical process designs have been proposed and discussed for more than twenty-five years, and have now been developed to the point of industrial application. Process synthesis has also now become an integral part of many chemical engineering design curricula. CACHE has contributed to the promotion and advancement of conceptual design and process synthesis methods through case studies, educational software products, symposia, and the sponsorship of process synthesis sessions within the *Foundations of Computer-Aided Process Design* conference series. Although much progress has been made, continuing research advances suggest that significant improvements in industrial and educational conceptual design and process synthesis methods and tools can be expected in the near future.

Introduction

When CACHE was established in 1969, design was the one aspect of chemical engineering that had most readily incorporated digital computing into the curriculum. Algorithms and software were becoming widely used for both the design of individual unit operations, and for the simulation of entire process flowsheets.

At that time, design was taught basically as a course in analysis. A flowsheet was given and students were asked to solve for the process material balances, energy balances, equipment sizes, utility flows, capital and operating costs, and the profitability of the process. The AIChE Student Contest problems and the Washington University Design Case Study Series followed this general format. Simulators were ideally suited to problems of this type. At this level of detail, more or less rigorous models were developed for the most expensive process units in order to optimize the design. The generation of process flowsheet alternatives themselves, however, received very little attention in design textbooks except to note that the activity requires creativity.

About 1970 academics became interested in the possibility of developing systematic procedures for inventing process flowsheets. This effort was called process synthesis because the

inputs to and outputs from the process were fixed, and the goal was to find flowsheets that could transform the inputs into the outputs. Preliminary analysis steps (often based on shortcut calculations) accompanied the synthesis steps to help make choices among alternative flowsheet structures. CACHE recognized the potential of this research and sponsored the first *Symposium* in the emerging field of computer-aided chemical process synthesis in 1972 (Seader, 1972).

Early Process Synthesis Explorations

Rudd (1968) suggested perhaps the first organized framework for design based on the notion of *systems decomposition.* He suggested that for any original design problem, a number of simpler specific subproblems should be identified the solutions to which when put together would plausibly solve the original problem. Typical chemical process design subproblems might include raw material receiving and storage, feed conditioning, reaction, product isolation, purification, storage, and shipping, and so on. Some sort of systems decomposition is a common conceptual design practice.

King and coworkers (1972) proposed an alternative *evolutionary modification* synthesis approach patterned on another industrial conceptual design practice. This approach starts with an existing flowsheet for the desired product or a similar product and then identifies aspects of the design that could be changed to greatest advantage to better meet the objectives of the specific case at hand as well as alternatives to implement that change. Various heuristic and analytical methods were proposed to identify evolutionary opportunities, but the success of the scheme depended strongly on the initial flowsheet.

As a third approach, Ichikawa and coworkers (1972) viewed process synthesis as an optimization over process structure and approached conceptual design through the application of operations research and mathematical programming techniques. This analysis-dominated approach starts with a larger superflowsheet which contains embedded within it many redundant alternatives and interconnections among them and then systematically strips the less desirable parts of the superstructure away. This *superstructure optimization* offers the promise of simultaneous optimization of structural as well as other design parameters. However, it requires a starting superstructure from somewhere (which for some simple problems, however, may be implicit in the formulation), as well as very extensive computational capability since the superstructure optimization problem is in general nonlinear, nonconvex, and involves both continuous and discrete variables.

Siirola (1971) suggested an alternative decomposition scheme based on an hierarchical ordering of physical properties. In the resulting *systematic generation* scheme, property differences between raw materials and desired products were detected and resolved using the *means-ends analysis* paradigm. The result was the application of chemical technologies in a sequence such that the raw materials become transformed into the desired products. The *property hierarchy* was molecular identity first, followed by species amount, concentration, phase, temperature, pressure, and possibly other form properties. Species identity differences were attacked first and resolved by the application of reaction technologies. Species amount differences were identified next and resolved by splitting or mixing or purchase. Concentration differences were identified next and generally resolved by separation technologies. Finally phase, temperature,

and pressure differences were detected and resolved by a variety of enthalpy-changing technologies. If a difference-resolving technology were identified but could not be directly applied to a given stream (for example, reaction that could not be applied because feeds were not pure enough, or a separation technology could not be applied because the feed stream was the wrong phase), recursive design subproblems were identified with the subgoal of producing stream conditions suitable for the application of the suspended technology, generally, but not always, by altering properties lower in the hierarchy. After all identified property differences were resolved, the property-changing tasks were integrated into actual processing equipment. The procedure may be repeated if desired at increasing levels of detail making use of screening results obtained at lower levels of detail.

Alternative solutions are generated when more than one technology is identified that can be applied to reduce or eliminate a property difference. The selection of which technology to choose might be made on the basis of rules or some evaluation at the time the technologies are being examined. Alternatively, each may be chosen separately and the consequences followed separately (leading to alternative design solutions) and then each final solution is evaluated. One additional possibility is that all feasible alternative technologies are selected and applied in parallel leading to a redundant design or superstructure. At the end of the design process, the superstructure is reevaluated in its entirety, and the less economical redundant portions eliminated.

The hierarchical decomposition paradigm became the basis of a series of flowsheet synthesis programs including AIDES (Siirola, 1970), BALTAZAR (Mahalec and Motard, 1977), and PIP (Kirkwood, Locke, and Douglas, 1988), as well as the Pinch Technology Onion (Linnhoff and Ahmad, 1983). It has also formed the basis for the present classification of the major process synthesis subproblems: Reaction Path Synthesis, Species Allocation (or Input-Output and Recycle Structure), Separation Scheme Synthesis, and Heat and Power Integration.

In 1990, the *PIP* (Process Invention Procedure) flowsheet synthesis software became available through CACHE. This software follows closely the hierarchical design procedure described by Douglas (1988) which involves alternating rule-based synthesis and short-cut analysis and economic evaluation which enables rough estimates of the optimum design. The method generates for each reaction step in a process, an input-output and recycle structure, a vapor recovery system for the reactor effluent, a liquid separation system for the reactor effluent, and a heat recovery network. Currently an expert system implementation of PIP is under development.

Heat Exchanger Network Synthesis

The pairing of complementary enthalpy-increasing tasks with enthalpy-decreasing tasks so as to optimally recover and reuse energy within a process while decreasing dependence on external utilities is known as *heat integration.* The synthesis of heat exchanger networks is the process synthesis subproblem that has received perhaps the most attention. This synthesis problem generally occurs late in the synthesis hierarchy after the various hot and cold streams have been fairly well defined with respect to amount, composition, available temperature, desired temperature, and other physical properties. Also, the tradeoffs between capital and operating cost are well understood.

It turns out that the maximum amount of heat integration and the minimum heating and cooling utility requirements can be determined for a process without actually designing a heat integration network through a technique called *Targeting*. Targeting procedures also exist for estimating the optimal number of exchangers and the total heat transfer area. These estimates can also be made in the light of constraints requiring or forbidding energy matches between specific pairs of streams. CACHE has made available for educational use one such program for calculating these estimates, *TARGET II*, developed by Linnhoff-March Inc.

Linnhoff and Hindmarsh (1983) popularized important representations including Composite Curves and Grand Composite Curves for visualizing the thermodynamics governing the heat exchanger network synthesis problem. They also introduced the concept of *Pinch Technology*, based on the point in the design where heat is transferred at the minimum approach temperature, which is especially useful in constraining the selection of energy matches which preserve overall minimum utility usage. Pinch technology also provides guidance on splitting streams for parallel heat exchangers, minimizing the total number of exchangers, proper incorporation of heat pumps, and integrating heat and power tasks. Within these constraints, procedures for the actual synthesis of heat exchanger networks are somewhat less well developed, but all of the standard process synthesis approaches including systematic generation, evolutionary modification, and superstructure optimization using all the standard tools of heuristics, expert systems, deterministic algorithms, and mathematical programming have been suggested. *THEN*, a heat exchanger matching algorithm developed by Carl Knopf, is available through CACHE. The straightforward application of heat integration technology to a typical process design generally reduces the net present cost on the order of 10%, due largely to sensible heat recovery.

As mentioned, heat exchanger networks have also been a testbed for superstructure optimization approaches to process synthesis. In this case, techniques from operations research have been used to essentially convert the synthesis problem into one of analysis that can be solved by mathematical programming. CACHE's sixth Process Design Case Study, *Chemical Engineering Optimization Models with GAMS*, contains three heat exchanger network synthesis examples including a linear programming transshipment model for predicting minimum utility consumption with stream matching constraints, a nonlinear programming model for global optimum search in the synthesis of heat exchanger networks, and a mixed-integer nonlinear programming model for simultaneous synthesis and optimization of heat exchanger networks.

Heat-Integrated Distillation System Synthesis

Synthesis methods for separation tasks depend in part on the nature of the system to be separated. In the case of distillative separations of relatively ideal systems, simple volatility ordering may be sufficient to represent the solution thermodynamics. Alternative separation trains can be synthesized using list processing and either heuristic rules (Seader and Westerberg, 1977) or dynamic programming techniques (Hendry and Hughes, 1972). If the number of components is not too large, all possible separation orderings may be generated and evaluated, as is implemented in the PIP software.

Distillation schemes involve a fair number of auxiliary enthalpy-changing tasks including reboilers, condensers, feed quality adjustment, and product coolers. Such distillation schemes

can be heat integrated with any of the standard techniques. However, since the temperatures of these enthalpy-changing tasks depend upon the pressures, purity specifications, and other operating conditions of each column in a distillation sequence, it is advantageous to simultaneously perform heat integration within the column pressure optimization (Siirola, 1981). Utility requirement reductions in the neighborhood of 50% and net present cost savings of 35% are typical for such simultaneous distillation sequence synthesis, operating parameter optimization, and heat integration. CACHE's sixth Process Design Case Study also includes an example of a GAMS-based MINLP approach for optimal sequencing and heat integration in the synthesis of sharp separation sequences.

For binary separations or processes that contain a single or a particularly dominant distillation, significant energy recovery by the previous approach may not be possible. However, a single distillative separation may be implemented as two columns in such a way that the latent heat rejected from one might be recycled to the other, approximately halving the net utility requirement. In order for this to be accomplished, the condenser of one column must be at a higher temperature than the reboiler of the other. This may be done by operating the columns at different pressures, possibly assisted by the degree of separation designed for each column. Blakely (1984) compared a number of such multiple effect designs against a single column as a function of feed flowrate, feed composition, and relative volatility. The specific results depend on the costs and temperatures of available utilities as well as on the capital cost correlations. In fact, at low flowrates the economics of the single column design are generally superior to all of the multiple effect designs largely because at those scales the capital costs of the columns dominate the energy costs. However at higher flowrates, for all conditions of feed composition and relative volatility, there exists one or more multiple effect designs with lower net present costs by as much as 35%, similar in magnitude to the savings achievable in the heat integration of other multiple distillation separations.

Nonideal Distillation Scheme Synthesis

For systems which exhibit more nonideal behavior involving azeotropes and regions of immiscibility, a more detailed representation of the thermodynamics may be necessary even for the synthesis of separation schemes at the targeting level of detail. One such representation is the *Residue Curve Map* developed by Doherty and coworkers (1985). Residue curve maps are derived from an analysis of a single-stage batch still, but may be applied to the understanding of the thermodynamic behavior of solutions and the behavior of continuous distillative separation operations on those systems. Residue curve maps define regions of feed compositions from which it is possible to reach the same lowest-boiling composition and the same highest-boiling composition by alternative operation of a single simple distillative separation task. Some systems involving azeotropes exhibit more than one such region separated by boundaries which normally can not be crossed by simple distillation. However techniques have been devised for overcoming these distillation boundaries including pressure shifting, exploiting boundary curvature, and exploiting separation technologies not involving volatility such as liquid-liquid decantation, solid-liquid crystallization, and kinetic phenomena such as membrane diffusion. The preparation of residue curve maps as well as the design of distillative separations consistent with these maps is being implemented in the MAYFLOWER software under devel-

opment at the University of Massachusetts.

An industrial separation synthesis procedure for systems exhibiting nonideal thermodynamics has been described in the recent CACHE-sponsored conference, *Foundations of Computer-Aided Process Design* (Siirola, 1994). Key features of the method include the *Residue Curve Map* representation of solution thermodynamics overlaid with pinched regions and liquid-liquid (or solid-liquid) equilibria data; identification of thermodynamic *Critical Features* to be avoided (pinched regions), overcome (distillation boundaries), or exploited (for example, liquid-liquid tie lines); a *Strategic Hierarchy* to address issues raised by the critical features first; the identification of *Interesting Compositions* (useful as mass separation agents which must be regenerated and recycled within the process); *Opportunistic Resolution* of remaining concentration property differences by mixing, decant, or distillative separation as permitted by distillation boundary constraints; and pursuit of *Multiple Alternatives* (heaviest underflow as well as lightest overhead) for every multicomponent distillative separation. The method also includes methods for pressure shifting distillation boundaries, exploiting boundary curvature, producing compositions which are neither the highest-boiling nor the lowest-boiling in a region, extractive distillation, handling recycle, removing redundancy, and heat integration. A similar rule-based system called SPLIT (Wahnschafft, Jurain, and Westerberg, 1991) is currently being incorporated into the ADVENT software by Aspen Technology, Inc. CACHE is working to make methods and tools such as these and other aspects of Residue Curve Map technology available for educational use.

Outlook for the Future

Many of the methods and tools being developed in the process synthesis research community are still more concerned with analysis (targeting, representation, operability, etc.) than with alternative generation. More effort will be directed specifically to alternative generation for conceptual design at all levels of detail.

Many of the applications discussed here were based on algorithmic systematic generation approaches or heuristic rule based or expert systems based approaches to process synthesis. Because of combinatorial difficulties, neither the algorithmic nor the heuristic methods carry any guarantees of structural optimality.

Superstructure approaches, on the other hand, while not especially practical at the present time, do offer with the promise of structural optimality and have tremendous potential for the future. As mentioned, several glimpses of how such approaches might be formulated are included in CACHE's sixth Process Design Case Study. There are significant challenges remaining to improve computational efficiency to overcome combinatorial difficulties and to develop global optimization strategies for the generally nonconvex mixed-integer nonlinear problems involved. Steady progress is being made on these fronts, and other process synthesis problems such as reaction path and reactor scheme synthesis, design for waste minimization, and consideration of other important design criteria such as safety, operability, and controllability. CACHE will continue its efforts to make advances in all of these process synthesis methods and tools available to the educational and industrial communities.

Conclusions

Systematic approaches to conceptual design and process synthesis have begun to have measurable industrial success. Although there are many restrictions, these techniques have led to higher value, lower energy, lower environmental impact, and sometimes even novel design alternatives. Process synthesis techniques have been successfully applied to the conceptual design of total process flowsheets, as well as to specific design subproblems including heat integration, heat-integrated distillation trains, multiple-effect distillation, and the separation of azeotropic systems. Typical energy savings of 50% and net present cost reductions of 35% have been achieved using systematic process synthesis methodologies.

Certain features of systematic approaches to process synthesis appear to have special merit. These include architectures which are hierarchical in scope and iterative in level of detail, targeting as a useful bounding exercise, means-ends analysis as a recursive problem-solving paradigm with hierarchical as well as opportunistic goals implemented by an iterative formulation-synthesis-analysis-evaluation design strategy, thinking specifically in terms of tasks to be accomplished before equipment to be employed, the importance of representations which encapsulate analysis, and the advantages of solving related synthesis problems simultaneously.

Advances in problem formulation and in computational hardware and software capability offer the promise of a new generation of practical process synthesis techniques based directly on structural optimization. Even greater benefits are expected to be realized as this next generation of approaches are transferred to industry and incorporated into design education. Soon the goal of synthesizing provably superior conceptual process flowsheets may be at hand.

References

Blakely, D.M. (1984). *Cost Savings in Binary Distillation through Two-Column Designs*. M.S. dissertation, Clemson University, Clemson.

Doherty, M.F. and G.A. Caldarola (1985), *Ind. Eng. Chem. Fundam.*, **24**, 474.

Douglas, J.M. (1988). *Conceptual Design of Chemical Processes*. McGraw-Hill, New York.

Foucher, E.R., M.F. Doherty, and M.F. Malone (1991). Automatic Screening of Entrainers for Homogeneous Azeotropic Distillation. *Ind. Eng. Chem. Res.*, **30**, 760.

Hendry, J.E. and R.R. Hughes (1972). Generating Separation Process Flowsheets. *Chem Eng. Progr.*, **68**(6), 69.

King, C.J., D.W. Gantz, and F.J. Barnes (1972). Systematic Evolutionary Process Synthesis. *Ind. Eng. Chem. Process Design Develop.*, **11**, 271.

Kirkwood, R.L., M.H. Locke, and J.M. Douglas (1988). A Prototype Expert System for Synthesizing Chemical Process Flowsheets. *Comput. Chem. Engng.*, **12**, 329.

Linnhoff, B. and S. Ahmad (1983). *Towards Total Process Synthesis.* AIChE Meeting paper 26d, Washington.

Linnhoff, B. and E. Hindmarsh (1983). The Pinch Design Method of Heat Exchanger Networks. *Chem. Eng. Sci*., **38**, 745.

Mahalec, V. and R.L. Motard (1977). Procedures for the Initial Design of Chemical Processing Systems. *Comput. Chem. Engng.*, **1**, 57.

Morari, M. and I. E. Grossmann, Eds (1991). *Chemical Engineering Optimization Models with GAMS*. CACHE Design Case Studies Volume 6, CACHE, Austin.

Rudd, D.F. (1968). The Synthesis of Process Designs, I. Elementary Decomposition Principle. *AIChE J*, **14**, 342.

Seader, J.D. (1972). *Computer-Aided Chemical Process Synthesis*. CACHE-Sponsored Symposium, CACHE, Cambridge.

Seader, J.D. and A.W. Westerberg (1977). A Combined Heuristic and Evolutionary Strategy for the Synthesis of Simple Separation Sequences. *AIChE J.*, **23**, 951.

Siirola, J.J. (1970). *The Computer-Aided Synthesis of Chemical Process Designs*. Ph.D. dissertation, University of Wisconsin, Madison.

Siirola, J.J. and D.F. Rudd (1971). Computer-Aided Synthesis of Chemical Process Designs. *Ind. Eng. Chem. Fundam.*, **10**, 353.

Siirola, J.J. (1981). Energy Integration in Separation Processes. In R.S.H. Mah and W.D. Seider (Eds.), *Foundations of Computer-Aided Process Design*, Vol. II. Engineering Foundation, New York. 573.

Siirola, J.J. (1994). An Industrial Perspective of Process Synthesis. In L.T. Biegler and M.F. Doherty (Eds.), *Foundations of Computer-Aided Design*. *AIChE Symposium Series*, Vol. 91, No. 304.

Umeda, T.A., A. Hirai, and A. Ichikawa (1972). Synthesis of Optimal Processing System by an Integrated Approach. *Chem. Eng. Sci.*, **27**, 795.

Wahnschafft, O.M., T.P. Jurain, and A.W. Westerberg (1991). SPLIT: A Separation Process Designer. *Comput. Chem. Engng.*, **15**, 656.

PROCESS SIMULATION

Lorenz T. Biegler
Carnegie Mellon University
Pittsburgh, PA 15213

J. D. Seader
University of Utah
Salt Lake City, UT 84124

Warren D. Seider
University of Pennsylvania
Philadelphia, PA 19104

Abstract

This brief discussion of process simulation is written from a standpoint of the impact of the contributions of CACHE on the educational community. Emphasis is on steady-state process simulation and optimization with FLOWTRAN. Some suggestions are made for needed future developments.

Introduction and Historical Development

One of the essential early steps in process design or analysis, following the synthesis of a process flowsheet and the establishment of a preliminary set of operating conditions suitable for a base-case study, is the computation of material and energy balances so as to determine: (1) for every major stream in the process, the flow rate, composition, temperature, pressure, and phase condition; and (2) for every major piece of equipment, the overall size and the shaft work or heat transfer requirements. Before the availability of digital computers, these calculations were done manually with a slide rule or mechanical desk calculator. If the process involved recycle streams, multistaged, multicomponent separation operations, and/or reactors with multiple reactions, the manual computations could be extremely tedious and time-consuming. Often, the computations would be done by approximate methods and incomplete closure of material balances was sometimes acceptable. If optimization studies were made, they were usually restricted to individual operations using one or two design variables, because a more global approach was not practical.

Two significant events occurred in the mid-1950s that were to drastically alter the approach to process simulation and design. The first was the introduction of the IBM 704 digital computer with its built-in floating-point arithmetic. The second was the development of the easy-to-use, high-level, procedure-based programming language, of J. W. Backus and others

from IBM, called FORTRAN. The IBM 704 came with a FORTRAN language compiler and subroutines written in that language were automatically handled. Almost overnight, chemical engineers in several large companies and in universities began writing large-scale FORTRAN computer programs to replace the manual methods. At first, these programs were called computer-aided, steady-state chemical process design programs. Later, they would be referred to as simply simulation or flowsheeting programs. The ability to replace tedious hand calculations by computerized calculations also spurred tremendous development in the areas of thermodynamic properties of liquids and gases, rigorous calculation of equilibrium-stage operations, and design of nonisothermal, multiple-reaction reactors, including effects of transport phenomena. At the same time, it was quickly realized that the digital computer was not limited to linear systems. Consequently, there was an explosion in the development of numerical methods for solving all sorts of nonlinear, as well as linear, problems.

By 1967, six commercial and two academic steady-state simulation programs, all written in FORTRAN, were in publicized use. Undoubtedly, other unpublicized commercial simulators were also in use. One of the widely publicized academic programs, PACER (Process Assembly Case Evaluator Routine), developed by Professor Paul T. Shannon at Purdue University, as reported in 1963, was made available to chemical engineering departments in 1967, following an instructional PACER workshop at Dartmouth College, April 23-28, 1967, which was attended by 20 chemical engineering professors from the United States and Canada. In 1971, the author of PACER together with Professors Crowe, Hamielec, Hoffman, Johnson, and Woods from McMaster University published the first book on simulation, entitled *Chemical Plant Simulation - An Introduction to Computer-Aided Steady- State Process Analysis*, which provided the necessary background to use PACER (Crowe, et al., 1971).

The academic version of PACER was mainly a modular executive program for directing material and energy calculations for flowsheets, which could include recycle streams. PACER had a limited library of unit-operation models and lacked a physical property estimation package. However, it was open-ended so that users could add operation and property models. In 1968, Professors R. L. Motard and Ernest J. Henley, at a workshop at the University of Houston that was attended by 42 professors, introduced a more advanced simulation program called CHESS (Chemical Engineering Simulation System). Like PACER, CHESS was based on a sequential modular architecture, in which models of the operating units were solved one at a time. For each unit, the inlet stream conditions were known or assumed and the outlet stream conditions and heat and work transfer rates were computed by the operation model. If the process did not involve recycle loops, the unit calculations began at the feed end of the process and progressed to the product end, so that the calculated outlet stream(s) from one unit became the inlet stream(s) to the next unit(s). If the process included one or more recycle loops, the user could provide a guess for the conditions of one stream in each loop and the program would automatically iterate each loop to convergence to some specified tolerance. The CHESS program included an expanded library of unit-operation models and a physical property package, making it easy to use. However, it lacked a general, rigorous, multistage, multicomponent separation unit for modeling distillation-type operations.

By the early 1970s, a number of new and mostly commercial steady-state simulation programs that contained rigorous absorption/stripping, distillation, and liquid-liquid extraction models began to emerge. The CACHE (Computer Aids for Chemical Engineering Education)

Committee, which had been formed in 1969, directed its Large-Scale Systems Task Force, chaired by Professor J. D. Seader, and the Program Distribution Task Force, chaired by Professor Warren D. Seider, to conduct a survey of academic and commercial simulation programs that might be made available at a reasonable cost for use by chemical engineering faculty and students. That survey identified a number of such programs and these two Task Forces then evaluated the programs with a comprehensive test problem that determined their ease of use and robustness. The FLOWTRAN program of Monsanto Company emerged as the preferred program.

FLOWTRAN was conceived in 1961 by the Applied Mathematics Department of Monsanto Company when they, like many large chemical processing companies, began contemplating the development of a generalized modeling computer program for process design and simulation. The project gained considerable impetus late in 1964 when Robert (Bob) H. Cavett, a brother of the entertainer Dick Cavett, left Pure Oil Company and joined the Applied Mathematics Department of Monsanto Company. While at Pure Oil, Bob Cavett had developed a physical property package for vapor-liquid systems and had investigated numerical analysis procedures for converging nonlinear systems of equations and recycle loops. Monsanto purchased from Pure Oil all rights to Cavett's work. By 1965, the name FLOWTRAN had been selected for the name of Monsanto's program, which was to be based on the sequential modular architecture. In 1966, the initial version of the program was available for use by chemical engineers at Monsanto. From 1969 to 1973, more than 70 outside companies used FLOWTRAN through Monsanto Enviro-Chem Systems, Inc., via commercial computer networks. However, this type of outside use was terminated by Monsanto in 1974 in favor of licensing the program.

On June 13, 1973, in a letter to Mr. John W. Hanley, President of Monsanto Company, CACHE requested that Monsanto consider allowing chemical engineering faculty and students to use FLOWTRAN in course work and research. With the assistance of Professor John J. McKetta of the Advisory Committee of CACHE and F. E. Reese, M. C. Throdahl, J. R. Fair, and S. I. Proctor of Monsanto, approval of the use of FLOWTRAN, via a national computer network, was granted by Monsanto Company on December 10, 1973. This approval included assistance in implementation of the system by providing a grant and loaning specialists from the Monsanto Corporate Engineering Department. Starting on May 10, 1974, Dr. Allen C. Pauls of Monsanto directed a three-day training course on FLOWTRAN, which was attended by Richard R. Hughes, H. Peter Hutchison, J. D. Seader, Warren D. Seider, and Arthur W. Westerberg. On June 5, 1974, at a meeting of the CACHE trustees in Seven Springs, Pennsylvania, Dr. Proctor of Monsanto formally presented to CACHE the Monsanto grant and conditions for making FLOWTRAN available to universities.

On July 1, 1974, with significant contributions from Pauls and Hughes, Seader and Seider completed the draft of a student textbook entitled *FLOWTRAN Simulation - An Introduction*. That book included material on the principles of simulation, detailed instructions on the application of FLOWTRAN to a wide variety of problems including a comprehensive design study by Hughes, numerous examples and homework problems, and descriptions of all of the equipment-unit and physical-property models. By August, 1974, this book was published by CACHE in a soft-bound edition and stocked for sale by Ulrich's Bookstore in Ann Arbor, Michigan.

The academic version of FLOWTRAN was almost identical to the commercial version that was used by Monsanto and licensed to other companies. The program consisted of 9 executive routines tailored to the particular computer on which the program was to run, 7 main FORTRAN programs, 2 data lists, 86 FORTRAN subroutines, and input data and output listings for 27 test problems that illustrated most of the features and capabilities of FLOWTRAN. The program was the result of 60 man-years of effort at a cost to Monsanto of more than two million dollars. At the time of the approval for university use, Monsanto engineers were running about 700 FLOWTRAN jobs per month. The application of FLOWTRAN to design and simulation was restricted to vapor-liquid systems. The library of operations models, called blocks, included 5 flash blocks, 5 distillation blocks, 2 absorber-stripper- extraction blocks, 8 heat exchanger blocks, 3 reactor blocks, 2 pump and compressor blocks, 5 stream mixing-dividing-manipulation blocks, 4 control blocks, 1 recycle convergence block, 16 equipment cost blocks, 4 financial analysis blocks, and 6 report blocks. Included in two of the flash blocks was the first algorithm for calculating a vapor-liquid-liquid (three-phase) flash, which was described in the 1969 textbook, "Material and Energy Balance Computations" by Professor E. J. Henley and Dr. E. M. Rosen of Monsanto Company. The control blocks allowed the user to manipulate certain equipment unit parameters to achieve desired values of stream variables associated with that unit or at another upstream unit.

With respect to thermodynamic properties, the academic version of FLOWTRAN included a property-constant data bank for 180 chemical species. Mixture property models were provided to estimate density, enthalpy, and K-values using the gamma-phi formulation. The Redlich-Kwong equation of state was used for non-ideality in the vapor phase and the equations of Scatchard-Hildebrand (regular solutions), van Laar, Wilson, and Renon-Prausnitz (NRTL) were available for non-ideality in the liquid phase. The latter three equations required binary interaction parameters that had to be supplied by the user. A special program, called VLE, included in the FLOWTRAN system, could be used to compute these parameters from experimental phase equilibria data. For species not in the data bank, the user could add their property constants to a private file. If the constants were not available, they could be estimated with a special program, called PROPTY, which was also included in the FLOWTRAN system.

When the FLOWTRAN program was being conceived, considerable effort was directed to the choice of architecture. With respect to the handling of recycle streams, both the sequential modular and simultaneous modular (two-tier) methods were debated. Although the latter was preferable for processes with a large number of recycle loops, the former was selected because it could be more readily understood and appeared to be more efficient on a number of test problems. A very flexible recycle convergence block was written, which could converge up to three recycle loops simultaneously and provided a number of options to the user, including the methods of successive substitution, bounded Wegstein, delayed Wegstein, and damped Wegstein. The user was required to specify the location of the convergence block in the recycle loop(s). A novel feature of the FLOWTRAN architecture was its variable internal structure. User input consisted of both data and commands in the form of ordered keywords pertaining to selection of chemical species and thermodynamic property models, and equipment unit blocks. Also, the user could insert FORTRAN statements among the block commands. The input was parsed into data and commands. The data were stored and the commands were translated into a main FORTRAN program, consisting mainly of block calls, which was then

compiled and linked to just those subroutines that were needed for that particular simulation. This type of architecture made FLOWTRAN efficient in mass storage requirements and open-ended so that user-supplied equipment unit subroutines could be readily added.

On August 13, 1974, Hughes, Seader, and Seider, with Dr. Pauls and Professor Richard S. H. Mah, held, at Northwestern University, the first of three three-day FLOWTRAN workshops, sponsored by the AIChE. The workshop was attended by 38 faculty members representing 35 different universities in the United States and Canada. One attendee remarked that FLOWTRAN was destined to alter the way process design was being taught to senior students in chemical engineering. Today, 20 years later, that prediction has come true; most, if not all, chemical engineering departments now teach computer-aided process design with any of a number of available steady-state process simulators.

The 1974 edition of *FLOWTRAN Simulation - An Introduction* was followed by a second edition in February, 1977, which added a few subroutines and a new chapter on writing cost blocks. A major revision was published in May, 1987, which included a new SQP optimization capability, a new Broyden-based convergence block, and a new chapter on optimization, all by Professor L. T. Biegler, as discussed in the section on optimization below. In January, 1975, CACHE published the booklet *CACHE Use of FLOWTRAN on UCS* by Hughes. This booklet, which was revised in January, 1982 with the assistance of Biegler, provided all the instructions necessary to access the FLOWTRAN program over the United Computing Systems (UCS) network, with computers located in Kansas City, Missouri. In December, 1977, CACHE published, *Exercises in Process Simulation using FLOWTRAN*, edited by Dr. J. Peter Clark. This book, which was revised in December, 1980, with the assistance of Dr. T. P. Koehler and Professor Jude T. Sommerfeld, contained problem statements and solutions for 8 short exercises, 14 workshop exercises, and 5 more comprehensive problems.

By mid-1975, one year after the publication of the FLOWTRAN textbook and the completion of the faculty workshop at Northwestern University, 25 universities were using FLOWTRAN. During the period from 1974 to 1983, an average of 21 universities/year were using FLOWTRAN on the UCS network. By 1985, 59 universities in the United States and Canada had used it. This represented about one-third of all the departments of chemical engineering.

Major deterrents to a wider use of FLOWTRAN were: (1) the inconvenience of having to submit card decks of input data in a batch mode to a remote batch machine, (2) the turn-around time for a run, which was typically one-to-two-days from the time the student submitted a card deck to the time the student received a print out, and (3) the cost of accessing the program on the UCS network. Typically a small-to-moderate size run would cost from $2.00 to $10.00. If access was from a remote job entry device, the cost would be more. Rarely did a student prepare an error-free input on the first, or even the second, try. Therefore, it might take a week to get a successful run. On the average, each university using FLOWTRAN was spending $1,500 per year.

In response to a request from CACHE, Monsanto Company announced on August 19, 1982 that load modules of FLOWTRAN on magnetic media would be made available to universities worldwide through CACHE for installation on in-house computers. The use of FLOWTRAN on the UCS network would be discontinued on January 15, 1986. By 1993,

FLOWTRAN load modules had been prepared for 14 different computer and operating system combinations, 190 (141 USA, 11 Canada, and 38 for 21 other foreign countries) load modules had been distributed worldwide, and approximately 15,000 copies of the three editions of *FLOWTRAN Simulation - An Introduction* had been sold.

Current Status of Steady-State Simulation

By the late 1980s, as discussed by Biegler (1989), computer-aided chemical process simulation had become virtually indispensable in process design and analysis. A large number of commercial simulators had become readily available and some of them, notably ASPEN PLUS of Aspen Technology, Inc., in Cambridge, Massachusetts, ChemCAD of Chem-Stations, Inc., in Houston, Texas, DESIGN II of ChemShare Corp. in Houston, Texas, HYSIM of Hyprotech Ltd. in Calgary, Alberta, Canada, and PRO/II of Simulation Sciences Inc., in Brea, California, were being licensed at low cost to universities. Compared to FLOWTRAN, some, most, or all of the latest versions of these simulators: (1) can be installed on a wider variety of computer/operating systems, including PCs with sufficient main and mass memory running under MS- or PC-DOS or one or more versions of Windows, (2) have much larger data banks, with more than 1000 chemical species, (3) have a much larger library of thermo-dynamic property models, including the ability to estimate K-values from equations of state, (4) have physical property models for entropy, transport properties, and other properties used to characterize materials, (5) have built-in tables for binary interaction parameters of activity coefficient correlations, (6) allow the user to specify any system of input and output units, (7) can generate graphs, (8) can handle solids, (9) have improved algorithms for converging multicomponent, multistage, separations, (10) have models for interlinked three-phase, and reactive distillation, (11) have models for kinetically controlled reactors, and (12) can compute simultaneous chemical and phase equilibrium by minimization of free energy. Thus, on April 4, 1994, Monsanto Company announced that it was their desire to bring to an orderly conclusion the CACHE FLOWTRAN project. Although FLOWTRAN can continue to be used for educational purposes on existing computers, no more load modules will be prepared or distributed. In 1993, at the AIChE Annual Meeting in St. Louis, the CACHE Corporation presented an award to Monsanto Company in recognition of its significant contribution to chemical engineering education. FLOWTRAN ushered in a major change in the way process design is taught to chemical engineers. More than any other individual, Robert H. Cavett of Monsanto was responsible for the successful development of sequential modular process simulation, which he implemented in FLOWTRAN. For this achievement, in 1987, he was the first recipient of the Computing Practice Award of the CAST Division of AIChE.

Optimization in FLOWTRAN

With the availability of FLOWTRAN from Monsanto in 1974, there was early interest by the late Professor Richard R. Hughes to also demonstrate and implement an optimization capability with process simulation and design. Professor Hughes had directed the development of the CHEOPS optimization package for Shell Development's process simulator and a similar capability was envisioned in the mid-1970s for FLOWTRAN. Fundamental work by Wilson and Beale in the mid-1960s led to Quadratic Approximation Programming (QAP), which was prototyped by Hughes and Isaacson in 1975. With QAP, reduced quadratic models for the ob-

jective function and constraints were constructed from the rigorous flowsheet models, and these reduced models were optimized to suggest a new base point. As a sequel to this work, A. L. Parker and Dick Hughes proposed an implementation of this optimization strategy to Monsanto, and a working version of QAP was demonstrated with an ammonia synthesis project using Monsanto's version of FLOWTRAN. Using about 65 simulation time equivalents on the ammonia process, this version took advantage of FLOWTRAN's data structures as well as the existing keyword interface. Still, this was the only problem solved by QAP on FLOWTRAN (Parker and Hughes, 1981).

Because the CACHE version of FLOWTRAN was available only through the UCS network, it took some time to implement Parker's approach for that version. As part of this exercise, a similarity between QAP and the evolving Successive Quadratic Programming (SQP) algorithm was noted, and this led to a simplified optimization algorithm which was termed Quadratic/Linear Approximation Programming (Q/LAP) (Biegler and Hughes, 1981). This approach was actually an SQP algorithm in disguise and improved the performance fourfold over the QAP strategy. However, developing an optimization approach proved expensive and clumsy on the UCS network and further development was done with a simpler prototype simulator, called SPAD, at the University of Wisconsin.

The QAP, Q/LAP, and virtually all previous flowsheet optimization approaches shared the same disadvantages. First, they required full flowsheet evaluations for objective and constraint function evaluations. This was expensive and also potentially unreliable; difficult intermediate points can lead to convergence failures. Second, these optimization approaches were implemented so that they needed to control the information flow in the overall process calculation. This led to a number of difficulties that deterred implementation of optimization strategies on existing process simulators.

However, the application of the SQP algorithm overcame both of these draw-backs. This algorithm can be interpreted both as the optimization of a quadratic model at each basepoint (iteration) as well as a (quasi-) Newton method applied to the optimality conditions of the process optimization problem. This second interpretation leads to the insight that both equality and inequality constraints could be converged simultaneously with a systematic improvement of the objective function. In particular, convergence of tear constraints, which often represented the least efficient part of a flowsheet simulation, could be incorporated into the formulation of the optimization problem and handled directly by the SQP algorithm. This insight further leads to a simplified implementation of an optimization algorithm to existing process simulators. Virtually all process simulators (and most prominently FLOWTRAN) have convergence blocks that appear as artificial units in the flowsheet calculation sequence. Upon convergence, a message is passed to the executive program of the simulator that signals the next calculation phase. This leads to the inclusion of sophisticated convergence algorithms largely independent of the architecture of the simulator and also allows the simple addition of an optimization capability - it is simply an extended convergence block. Moreover, this flexibility appears to have been anticipated by early SQP codes as well (e.g., VF02AD in the Harwell library). These were structured through reverse communication and thus allowed the information flow to be controlled by the user or a comprehensive modeling package. It appears that both recycle convergence blocks and the SQP code were made for each other. Not surprisingly, the insights of linking SQP with process simulators was also discovered independently at about the same time at the University

of Illinois by Chen and Stadtherr (1985) and at Cambridge University by Kaijaluoto (1985).

At the University of Wisconsin and later at Carnegie-Mellon University, this optimization approach was demonstrated with the SPAD simulator by Biegler and Hughes (1982) and Ganesh and Biegler (1987) on several moderately-sized process flowsheets. However, SPAD proved to be limiting for further development and demonstration of flowsheet optimization algorithms. Fortunately, by 1984, FLOWTRAN load modules had been developed and distributed throughout academia for instructional use. A key feature of FLOWTRAN was its run-time compile architecture, which allowed a powerful FORTRAN insert capability both in specifying the problem and in linking user-written models. (This feature has also been used to advantage in the ASPEN simulator.) Moreover, the instructional documentation encouraged the development of these models by providing clear and ample information on data structures and internal utility routines. As a result, it was natural to reconsider FLOWTRAN as a candidate for demonstrating advanced flowsheet optimization strategies.

The SCOPT (Simultaneous Convergence and Optimization) package combined an improved SQP algorithm along with a tailored interface to the FLOWTRAN simulator. Written as a recycle convergence block with access to global decision variables and other FLOWTRAN data structures, it was easily implemented as a NEW BLOCK and required virtually no restructuring of the existing simulator. Moreover, existing FLOWTRAN features could be used directly in an efficient manner. Optimization problems were formulated through in-line FORTRAN and decision and dependent variables were identified through PUT statements, as detailed in the FLOWTRAN documentation. As a result, even a casual FLOWTRAN user could easily set up and optimize a given flowsheet.

The SCOPT package was written largely at Carnegie-Mellon with generous advice and assistance from the FLOWTRAN group at Monsanto. Between 1984 and 1986, the code was tested and documented, and more than a dozen test problems were developed and benchmarked. These included several unit and small loop optimizations, along with flowsheet optimization for ammonia synthesis, methanol synthesis, allyl chloride manufacture, and monochlorobenzene production. In addition, beta testing was carried out at a number of universities including the University of Pennsylvania, the University of Utah, and the University of Wisconsin. As a result, the CACHE FLOWTRAN simulator was fully extended with an optimization capability, and in 1986 the SCOPT block became an integral component of the FLOWTRAN package.

As part of FLOWTRAN, SCOPT has found considerable use in senior and graduate level courses in process simulation, design and optimization. Many have used this capability, for instance, in open-ended design projects. In addition, the SCOPT package has spawned a significant body of graduate research. At Carnegie-Mellon, the development of advanced optimization algorithms was done with SCOPT by Lang and Biegler (1987), along with the analysis and demonstration of simultaneous strategies for heat integration and flowsheet optimization by Lang, Biegler, and Grossmann (1988). Moreover, an early version of mixed integer nonlinear programming (MINLP) for process retrofits was also prototyped with SCOPT by Harsh, Saderne, and Biegler (1989). In addition, the SQP code from this package has been used separately to solve nonlinear programming problems in a variety of research domains. The SCOPT package has been adapted and modified for a number of commercial and academic

simulation tools. For example, a modification of the SCOPT package forms the optimization capability in the SEPSIM process simulator of the Technical University of Denmark.

As FLOWTRAN has been superseded by more powerful commercial simulators, the classroom use of flowsheet optimization continues to evolve. While SQP still remains the workhorse of flowsheet optimization tools, numerous improvements have been made to this algorithm over the last decade. Work on incorporating SQP optimization strategies in commercial flowsheet simulators was started in the mid-80s. Most of these simulators, such as ASPEN PLUS, DESIGN/II, and PROCESS (later PRO/II), apply a structure similar to the one in FLOWTRAN. As with FLOWTRAN, probably the greatest advantages to these optimization approaches are the fast convergence of SQP (relatively few flowsheet evaluations are required) and the ease with which the optimization strategy can be implemented in an existing simulator. In fact, a further step was taken with the ProSim simulator, developed in France, where analytical derivatives were supplied for each module. SQP flowsheet optimization resulted in an order of magnitude improvement in performance with higher quality solutions!

Recently, there has also been strong development in equation-based process simulation, especially for optimization of process operations in real-time. A number of equation-based modeling tools (including SPEEDUP, DMO, MASSBAL, and NOVA) have been developed and all of these rely on SQP optimization strategies that are adapted to large-scale problems. Compared to the modular approaches of most sequential-modular simulators, as described above, equation-based strategies can be more than an order of magnitude faster, since all of the flowsheet equations are now converged simultaneously. However, equation-based simulators lack some of the intuitive structure of modular approaches, making them harder to construct and use. To preserve this modular structure and still effect simultaneous convergence of all the equations, the SQP strategy has recently been adapted (Schmid and Biegler, 1994) to deal with existing modules (e.g., columns and reactors) that converge simultaneously with the optimization problem and thus obtain the best of both worlds.

An important limitation of the flowsheet optimization method described above is that it can only find local optima. Therefore, good initializations are required to yield “good” (if not globally optimal) solutions. Recently, deterministic optimization algorithms have been developed that guarantee convergence to global optima for a large class of nonlinear problems. These strategies are somewhat more expensive than local tools and rely on combinatorial searches and convex relaxation in order to develop upper and lower bounds to the global solution. Often, they also incorporate local optimization solvers as well. A comprehensive review by Floudas and Grossmann (1994) summarizes their performance on structured process problems, such as pooling problems and heat exchanger networks. While, global optimization methods have yet to address problems of the size and generality of flowsheet optimization, progress in the last five years has been significant and is enhancing the industrial perception and application of these optimization tools. It is certain that global optimization and other future advances will be made in the development of flowsheet optimization algorithms and in the proper formulation of such problems. However, judging from the FLOWTRAN experience, it is crucial that these advances be aided by the development of widely distributed and user-friendly software. The resulting advice, support, and feedback are essential in identifying and tackling conceptual challenges and in opening up new areas for future research.

The Future of Process Simulation

Despite the significant progress that has been made in process simulation since the 1960s, users of computer-aided simulators are well aware of lingering limitations that still require remedies. As discussed by Seider, Brengel, and Widagdo (1991), current simulators are not yet utilizing the many new mathematical tools and approaches that are readily available. New numerical analysis tools such as homotopy continuation and the interval-Newton method with generalized bisection are needed to improve the robustness of nonlinear-equation solution, especially for multicomponent, multistage, equilibrium-based separation models, which for the more difficult problems fail to converge all too frequently. Currently, a simulator only searches for one solution. New tools, such as continuation and bifurcation, with graphical displays, are needed for assisting in the determination of solution multiplicity. Dynamic simulators are just beginning to emerge. They offer the opportunity to design more efficient and optimal control systems. Commercial rate-based separation operation models are becoming available and should be widely used in the near future. Membrane, adsorption, ion-exchange, chromatography, crystallization, and drying models need to be added to unit operations libraries. Equilibrium calculations need to be extended to handle solid phases, as well as liquid and gas phases.

References

Biegler, L. T., and R. R. Hughes (1981). *Chem. Eng. Progress*, **77**(4), 76.

Biegler, L. T., and R. R. Hughes (1982). *AIChE J.*, **28**, 994.

Biegler, L. T. (1989). *Chem. Eng. Progress*, **85**(10), 50-66.

Chen, H-S., and M. A. Stadtherr (1985). *AIChE J.*, **31**, 1843.

Clark, J. P., T. P. Kohler, and J. T. Sommerfeld (1980). Exercises in Process Simulation Using FLOWTRAN, Sec. Ed., *CACHE*.

Crowe, C. M., A. E. Hamielec, T. W. Hoffman, A. I. Johnson, D. R. Woods, and P. T. Shannon (1971). *Chemical Plant Simulation*, Prentice-Hall, New Jersey.

Floudas, C. A., and I. E. Grossmann (1994). *Proc. FOCAPD*, Snowmass, CO.

Ganesh, N., and L. T. Biegler (1987). *AIChE J.*, **33**, 282.

Harsh, M. G., P. Saderne, and L. T. Biegler (1989). *Comp. Chem. Eng*., **13**, 947.

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

Hughes, R. R. (1982). CACHE Use of FLOWTRAN on UCS, Sec. Ed., *CACHE*.

Kaijaluoto, S., (1985). PhD Thesis, Cambridge University.

Lang, Y.-D., and L. T. Biegler (1987). *Comp. Chem. Eng*., **11**, 143.

Lang, Y.-D., L. T. Biegler, and I. E. Grossmann (1988). *Comp. Chem. Eng*., **12**, 311.

Parker, A. L., and R. R. Hughes (1981). *Comp. Chem. Eng*., **5**, 123.

Schmid, C., and L. T. Biegler (1994). *Trans. I. Chem. E*., **72**, 382.

Seader, J. D., W. D. Seider, and A. C. Pauls (1987). FLOWTRAN Simulation - An Introduction, Third Ed., *CACHE*.

Seider, W. D., D. D. Brengel, and S. Widagdo (1991). *AIChE J*., **37**, 1-38.

OPTIMIZATION

Ignacio E. Grossmann
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This chapter presents a general classification of optimization models, and discusses the role that several modelling systems can play in education for helping students to learn problem and model formulation, and to address relevant problems in chemical engineering.

Introduction

Chemical engineering is a very fertile area for optimization studies, since many problems can be formulated as mathematical models in which an objective function is to be minimized or maximized subject to a set of equality and/or inequality constraints. For example, the optimal operation of a chemical process can be modelled with an objective function of profit that is to be maximized and that accounts for the revenues of the product and cost of raw materials and utilities. The equality constraints (i.e., equations) correspond to the mass and heat balances, and the inequalities to process specifications (e.g. product purity) and physical limits (e.g. non-negative flows, maximum pressure). In the past the teaching of optimization was largely an interesting theoretical exercise, but of limited application. The main emphasis was placed on one variable optimization problems and on linear programming. Most of the problems in chemical engineering, however, are multivariable and nonlinear, with constraints, and often require the use of discrete valued variables. Also, while optimization is a conceptually powerful framework, its impact has been somewhat limited in practice due to the lack of tools that facilitate the formulation and solution of these problems. In fact the common practice has been to rely on callable FORTRAN routines.

Recent years, however, have seen the development not only of new optimization algorithms, but also of powerful modelling systems. The important feature of these systems is that they provide an environment where the student need not be concerned with the details of providing the interfaces with various optimization codes (e.g. calling the appropriate routines, supplying the correct arguments for variables and functions and their derivatives, etc.). Instead, this environment automates the interfacing to a variety of algorithms that allow the student to *concentrate on the modelling of problems*, ultimately the main skill required for successful application of optimization in practice. More importantly, such an environment greatly reduces the time - up to one order of magnitude - required for the student to formulate and solve an op-

timization problem. This in turn allows the student to solve considerably more complex and interesting optimization problems than has been the case in the past.

We first give a brief overview of optimization models. We then introduce several modelling systems and educational materials that have proved to be useful in undergraduate and graduate education. Finally, we close with a discussion on future trends of the teaching of optimization.

Overview of Optimization

The use of optimization models in chemical engineering arises in a large variety of applications such as process design, process operation, process control, data fitting, and process analysis. Examples in process design include the synthesis of and sizing equipment for process flowsheets, separation sequences, or heat exchanger networks. Examples in process operations include production planning for refineries and production scheduling of batch processes. In process control, a good example is the determination of optimal temperature profiles in plug flow reactors. Data fitting involves both linear and nonlinear regression. An example of process analysis is the chemical equilibrium calculation through the minimization of the Gibbs free energy. All these problems have in common that they give rise to mathematical programming models involving continuous and discrete variables. These variables must be selected to satisfy equations and inequality constraints while at the same time optimizing a given objective function; the model is:

$$\begin{aligned} & \min f(x,y) \\ \text{s.t.} \quad & h(x,y) = 0 \\ & g(x,y) \le 0 \\ & x \in X,\ y \in Y \end{aligned} \qquad \text{(MP)}$$

The continuous variables are represented by x and the discrete variables by y, both with arbitrary dimensionality. The feasible region for the variables x and y is defined by the following constraints: $h(x,y) = 0$, equations describing the performance of a system (e.g., mass and energy balances, design equations); $g(x,y) \le 0$, inequalities related to specifications (e.g., minimum purity, maximum throughput), and by the sets X and Y. The former typically specify the ranges of values for the continuous variables (e.g., positive flows, minimum/maximum pressures for a chemical reactor), while the latter specifies the discrete choices (e.g. only 0-1 choices for say selection or not of a unit, or an integer number for number of plates). It should be noted that the model in (MP) can readily accommodate maximization problems (equivalent to minimizing the negative of the objective function), or problems involving greater than or equal to zero inequalities.

The mathematical model in (MP) is technically known as a “mathematical program” and an extensive body of literature exists on this subject (see Nemhauser et al., 1989, for a general review). From a practical standpoint, the important feature of a model MP is that it provides a powerful framework for modeling many optimization problems. Depending on the application at hand and the level of detail in the equations, the objective function and constraints can be given in explicit or implicit form. Explicit equations are often used for simplified models that

are expressed with algebraic equations. In this case the evaluation of the model may be cheap, but may involve thousands of variables and constraints. Implicit equations arise when detailed and complex calculations must be performed in “black boxes”, the case, for instance, with process simulator or differential reactor models. In these cases, the dimensionality for the optimization can be greatly reduced, but the evaluation of trial points for the optimization may be very expensive.

Formulating a given decision problem as a mathematical program requires three major assumptions: (a) that the criterion for optimal selection can be expressed through a single objective function, (b) that all the constraints must be exactly satisfied, and (c) that the parameters are deterministic in nature. Of course, these represent oversimplifications of real world problems. However, it should be noted that extensions based on the model in (MP) are available for addressing some of these issues. For instance, it is possible to relax the assumptions in (a) and (b) through multiobjective optimization methods, while the assumption in (c) can be relaxed through stochastic optimization methods.

If we assume no special structure to the problem in MP and to its various particular cases (e.g., only discrete or only continuous variables), then direct search techniques are often the easiest, but also the most time consuming, methods to apply. Here, either a systematic or random selection of trial points is chosen to evaluate and improve the objective function. Satisfaction of constraints can also be accomplished, but frequently with some difficulty (e.g., by using penalty functions). Perhaps the most popular direct search method that has emerged recently in chemical engineering is simulated annealing. This method is based on analogies with free energy minimization in statistical mechanics. This method is in principle easy to apply to problems with simple constraints, and is likely to find solutions that are close to the global optimum. However, aside from the fact that it often requires many thousands of function evaluations before the likely optimum is found, its performance tends to be highly dependent on the selection of parameters of the algorithm.

On the other hand, the most prevalent approach taken to date in optimization is to consider particular problem classes of MP, depending on the form of the objective function and constraints for which efficient solution methods can be derived to exploit special structures. The best known case of MP is the linear programming (LP) problem in which the objective function and all the constraints are linear, and all the variables are continuous:

$$\begin{aligned} & \min Z = c^T x \\ \text{s.t.} \quad & A\,x \leqq a \\ & x \geq 0 \end{aligned} \qquad \text{(LP)}$$

LP problems have the property that the optimum solution lies at a vertex of the feasible space. Also, any local solution corresponds to the global optimum. These problems have been successfully solved for many years with computer codes that are based on the simplex algorithm. Major changes that have taken place over the last ten years at the level of solution methods are the development of interior point algorithms that rely on nonlinear transformations and whose computational requirements are theoretically bounded by a polynomial expressed in terms of the problem size. Interestingly, this property is not shared by the simplex algorithm, which theoretically may require exponential time. Since this performance is rarely observed in

practice, further significant advances have taken place for solving large scale problems with the simplex algorithm. With this algorithm, problems with up to 15,000 to 20,000 constraints can be solved quite efficiently; interior point methods tend to perform better in problems with up to 50,000 to 100,000 constraints.

The extension of the LP model that involves discrete variables is known as a mixed-integer linear program (MILP). The most common case is when the discrete variables have 0 or 1 values:

$$\begin{aligned} & \min Z = a^T y + c^T x \\ \text{s.t.} \quad & By + Ax \leqq b \\ & y \quad \{0,1\}^m \quad x \geq 0 \end{aligned} \qquad \text{(MILP)}$$

This model greatly expands the capabilities of formulating real world problems, since one can include logical decisions with 0-1 variables, or, in general, account for discrete amounts. The most common method for MILP is the branch and bound search, which consists of solving a subset of LP subproblems while searching within a decision tree of the discrete variables. The other common approach relies on the use of cutting planes that attempt to make the MILP solvable as an LP with the addition of constraints.Because of the combinatorial problem that is introduced by the discrete variables, MILP problems have proved to be very hard to solve. In fact, theoretically, one can show that this class of problems is NP-complete; that is, there is no known algorithm for solving these problems in polynomial time. Nevertheless, recent advances based on combining branch and bound methods with cutting planes, and which have been coupled with advances in LP technology, are providing rigorous optimal solutions to problems that were regarded as unsolvable just ten years ago.

For the cases where all, or at least some, of the functions are nonlinear, and only continuous variables are involved, MP gives rise to nonlinear programming problems (NLP):

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \\ & x \quad R^n \end{aligned} \qquad \text{(NLP)}$$

For the case when the objective and constraint functions are differentiable, local optima can be defined by optimality conditions known as the Kuhn-Tucker conditions. These are perhaps the most common types of models that arise in chemical engineering. While ten years ago problems involving 100 variables for NLP were regarded as large, currently the solution of problems with several thousand variables is quite common. Reduced gradient and successive quadratic programming techniques, which can be derived by applying Newton's method to the Kuhn-Tucker conditions, have emerged as the major algorithms for NLP. The former method is better suited for problems with mostly linear constraints, while the latter is the method of choice for highly nonlinear problems. A limitation of these methods is that they are only guaranteed to converge to a local optimum. For problems that involve a convex objective function and a convex feasible region this is not a difficulty since these problems exhibit only one local

optimum which therefore to the global optimum.

The extension of nonlinear programming for handling discrete variables yields a mixed-integer nonlinear programming (MINLP) problem that in its general form is identical to problem (MP). The more specific form that is normally assumed is linear in 0-1 variables and nonlinear in the continuous variables,

$$\begin{aligned} &\min Z = c^T y + f(x) \\ &B\,y + g(x) \le 0 \qquad\qquad (\text{MINLP}) \\ &x \quad X = \{x \mid x \in R^n,\ x^L \le x \le x^U\} \\ &y \quad \{0,1\}^m \end{aligned}$$

MINLP problems were regarded as essentially unsolvable ten years ago. New algorithms such as the outer-approximation method and extensions of the Generalized Benders decomposition method have emerged as the major methods. These methods, which assume differentiability of the functions, consist of solving an alternating sequence of NLP subproblems and MILP master problems. The former optimize the continuous variables, while the latter optimize the discrete variables. As in the NLP case, global optimum solutions can only be guaranteed for convex problems. Solution of problems with typically up to 50 to 100 0-1 variables and 1000 continuous variables and constraints have been reported with these methods. Major difficulties encountered in MINLP include the ones encountered in MILP (combinatorial nature requiring large-scale computations) and in NLP (nonconvexities yielding local solutions).

Finally, it should also be noted that all the above methods assume that the problem in (MP) is expressed through algebraic equations. Very often, however, these models may involve differential equations as constraints. This gives rise to problems known as optimal control problems or as optimization of differential algebraic systems. The major approach that has emerged here is to transform these problems through discretization into nonlinear programming problems. The alternate approach is to solve the differential model in a routine which is then treated by the optimizer as a black box.

From the above review it is clear that there has been substantial progress in optimization algorithms over the last decade. These advances have in fact been accelerated by increased computational power and by the advent of powerful modeling systems, discussed in the next section.

Modelling Systems

Modelling systems have had a very large impact in practice, by making software systems in which optimization problems can be easily formulated in equation form readily accessible to many users of PC's, workstations and supercomputers. By separating the declarative part of a problem from the solution algorithms and their computer codes, these non-procedural modeling systems allow students to concentrate on the formulation of their models. These systems have typically reduced by at least one order of magnitude the time required to formulate optimization problems. For instance, in the past the equations for a model and any required analytical derivatives had to be supplied through subroutines or clumsy data files. This was not only

a time consuming process, but also one that was prone to producing many errors. With current modeling systems, many of these problems have been virtually eliminated, greatly facilitating the formulation and solution of optimization problems.

We will now briefly review three modelling systems that have proved to be useful in education.

LINDO. This is an interactive program for solving LP and MILP problems. The simplex algorithm is used for the LP problem and the branch and bound method is used for the MILP. The program is available in PC and Macintosh versions, as well as for the UNIX operating system. LINDO is especially suitable at the undergraduate level due to the great ease of its use. When specifying an LP or MILP model, any name can be used for the variables. Also, there is no need to indicate the product of a coefficient times a variable with the * sign. Inequality signs are <= for less than or equal to, and => for greater than or equal to. There is also no need to explicitly specify non-negativity constraints on the variables, since all variables are assumed to have values greater than or equal to zero.

The problem formulation for a very small LP can be entered interactively as shown below (input in bold):

Enter model

MAX X1 + 2 X2
? ST
? X1 + X2 <= 3
? END

Examine model

LOOK ALL

```
  [1] MAX    X1 + 2 X2
SUBJECT TO
  (2) X1 + X2 <=  3
END
```

Solve the LP

GO

Output from LINDO

```
 LP OPTIMUM FOUND AT STEP      1
      OBJECTIVE FUNCTION VALUE
      1)    6.00000000
 VARIABLE       VALUE              REDUCED COST
      X1         .  000000             1.000000
      X2         3.000000               .000000
     ROW   SLACK OR SURPLUS     DUAL PRICES
      2)           .000000             2.000000
```

The formulation can be edited and stored in a file. The model can also be specified in a file and accessed through the LINDO command TAKE. There are, of course, a number of other options which are explained in the program. The major advantages of LINDO are its great ease of use (no manual is needed) and its reliability. The instructor need spend almost no class time explaining the use of the program. A one page handout explaining the basics is sufficient. The only disadvantage of LINDO is that all equations must be entered in explicit form. Thus, one cannot use indexed variables and equations for specifying models in compact form. Therefore, LINDO is most suitable for small problems. The LINDO software and its book (Schrage, 1984) are distributed by LINDO Systems, Inc. The student version is available for problems with up to 100 constraints and 120 variables on the PC and 60 constraints and 120 variables on the Macintosh. Larger versions are also available.

GINO. This is an interactive program for solving NLP problems whose user interface is similar in nature to LINDO's. GINO uses a feasible path variant of the reduced gradient algorithm; the partial derivatives for the gradients are computed numerically by finite differences. The program is available in PC and Macintosh versions, as well as for the UNIX and VMS operating systems. GINO is suitable at the undergraduate level because of its great ease of use. Furthermore, GINO can also be used as an equation solver, and is also useful in courses such as thermodynamics and chemical reaction. Algebraic expressions used in the NLP can have arbitrary variable names, and the basic operator signs are as follows:

+ - Addition, subtraction

* Multiplication

/ Division

^ Exponentiation

LOG(X) natural logarithm

EXP(X) exponential function

= equal

<= less than or equal to

=> greater than or equal to

Each function in the formulation must end with a semicolon (;). The objective function is specified as MIN = or MAX = .

The problem formulation for the small NLP:

$$\begin{aligned} & \min Z = x^2 + y^2 \\ \text{s.t.} \quad & 2x + y \geq 4 \\ & x \geq 0 \qquad y \geq 0 \end{aligned}$$

can be entered interactively in GINO as shown below (input in bold):

Enter the model

```
MODEL:
? MIN= X ^ 2 + Y ^ 2 ;
? 2 * X + Y > 4 ;
? END
SLB X 0
SLB Y 0
```

Solve the NLP

```
GO
 SOLUTION STATUS: OPTIMAL TO TOLERANCES. DUAL CONDITIONS: SATISFIED.
        OBJECTIVE FUNCTION VALUE
     1)       3.200000
  VARIABLE       VALUE          REDUCED COST
       X          1.600000          0.000000
       Y          0.800000          0.000000
     ROW   SLACK OR SURPLUS      PRICE
     2)         -0.000000        -1.600001
```

The formulation can be edited and stored in a file, or it can be entered through an input file which is accessed with the GINO command take. As with LINDO, the major advantages of GINO are its great ease of use (no manual is needed) and reliability. Also, as mentioned above, GINO can be used as a nonlinear equation solver. The disadvantage of GINO is that no indexed equations can be used. Therefore, this program is most suitable for small problems. The GINO software and its book (Liebman et al. 1986) are distributed by LINDO Systems, Inc. The student version is available for problems with up to 30 constraints (excluding simple lower and upper bounds) and 50 variables on both the PC and Macintosh. Larger versions are available.

GAMS. GAMS is an advanced modelling system which is not as easy to use as LINDO and GINO, but provides a flexible interface for the formulation of LP, MILP, NP and MINLP models and their solution with a variety of different algorithms. The models are supplied by the user in an input file (Filename.GMS) in the form of algebraic equations using a higher level language. GAMS then compiles the model and interfaces automatically with a "solver" (i.e., optimization algorithm). The compiled model, as well as the solution found by the solver, are then reported back to the user through an output file (Filename.LST on a PC and UNIX workstation, Filename.LIS on VMS).

In order to compile and execute the input file, the command is simply: GAMS filename. The GAMS input file is in general organized into the following sections:

a. Specification of indices and data
b. Listing of names and types of variables and equations (constraints and objective function)
c. Definition of the equations (constraints and objective function).

d. Specification of bounds, initial values and special options.

e. Call to the optimization solver.

While the format of the input files is not rigid, the syntax is. Also, there is a rather large number of keywords that provide flexibility for handling simple and complex models (all in equation form, either explicitly or with indices).The main solver types available in GAMS are as follows:

LP	linear programming
NLP	nonlinear programming
MIP	mixed-integer linear programming
RMIP	relaxed MILP; the integer variables are treated as continuous.
MINL	mixed-integer nonlinear programming

The optimization software available in the student edition of GAMS by Scientific Press is as follows:

LP	ZOOM, MINOS
MIP, RMIP	ZOOM
NLP	MINOS

ZOOM is a simplex and branch and bound code for LP and MILP, while MINOS is an infeasible path implementation of the reduced gradient method. In addition, there is a special student edition available through CACHE (see section below) which also includes the following software:

MINLP	DICOPT++

DICOPT++ implements an augmented penalty variant of the outer-approximation method.

As an illustration of GAMS, consider the problem of assigning process streams to heat exchangers as described in pp. 409-410 of the text *Optimization of Chemical Processes* by Edgar and Himmelblau. The optimization problem is given by:

$$\begin{aligned} \min\ Z &= \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} &= 1 \qquad j = 1..n \\ \sum_{j=1}^{n} x_{ij} &= 1 \qquad i = 1..n \\ x_{ij} &= 0,1 \qquad i = 1,\ n \quad j = 1,\ n \end{aligned} \tag{AP}$$

which corresponds to the well known assignment problem. Here, i is the index for the n streams and j is the index for the n exchangers. The binary variable $x_{ij} = 1$ if stream i is assigned to exchanger j, and $x_{ij} = 0$ if it is not. The two equations simply state that every exchanger j must be assigned to one stream, and every stream i must be assigned to one exchanger.

The cost C_{ij} of assigning stream i to exchanger j is as follows:

		Exchangers			
Streams		1	2	3	4
	A	94	1	54	68
	B	74	10	88	82
	C	73	88	8	76
	D	11	74	81	21

We can formulate the above problem in GAMS almost in the form of the model in (AP) using index sets. The output file is shown below.

```
Test Problem
  4 *
  5 * Assignment problem for heat exchangers from pp.409-410 in
  6 * "Optimization of Chemical Processes" by Edgar and Himmelblau
  7 *
  8
  9  SETS
 10       I   streams      / A, B, C, D /
 11       J   exchangers   / 1*4 / ;
 12
 13  TABLE C(I,J)   Cost of assigning stream i to exchanger j
 14
 15           1     2     3     4
 16     A    94     1    54    68
 17     B    74    10    88    82
 18     C    73    88     8    76
 19     D    11    74    81    21 ;
 20
 21
 22  VARIABLES X(I,J), Z ;
 23  BINARY VARIABLES X(I,J);
 24
 25  EQUATIONS ASSI(J), ASSJ(I), OBJ;
 26
 27  ASSI(J)..   SUM( I, X(I,J) ) =E= 1;
 28  ASSJ(I)..   SUM( J, X(I,J) ) =E= 1;
 29  OBJ..      Z =E= SUM( (I,J), C(I,J)*X(I,J) ) ;
 30
 31  MODEL HEAT / ALL / ;
 32
 33  OPTION LIMROW = 0;
 34  OPTION LIMCOL = 0;
 35  OPTION MIP=ZOOM;
 36  OPTION SOLPRINT = OFF;
 37
 38  SOLVE HEAT USING MIP MINIMIZING Z;
 39
 40  DISPLAY X.L, Z.L ;

MODEL STATISTICS
BLOCKS OF EQUATIONS      3    SINGLE EQUATIONS       9
BLOCKS OF VARIABLES      2    SINGLE VARIABLES      17
NON ZERO ELEMENTS       49    DISCRETE VARIABLES    16
```

```
          S O L V E     S U M M A R Y
    MODEL   HEAT            OBJECTIVE Z
    TYPE    MIP             DIRECTION MINIMIZE
    SOLVER ZOOM             FROM LINE 38
**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE              97.0000
 RESOURCE USAGE, LIMIT         0.390     1000.000
 ITERATION COUNT, LIMIT        16        1000
                Iterations      Time
 Initial LP             16      0.06
 Heuristic               0      0.00
 Branch and bound        0      0.00
 Final LP                0      0.00

----    40 VARIABLE X.L
          1         2         3         4
A                                   1.000
B                 1.000
C                           1.000
D       1.000

----    40 VARIABLE Z.L                 =       97.000
```

Note that the SOLVE SUMMARY indicates that the optimum objective function value is Z = 97. Note also that the solution was obtained from the relaxed LP. This is not surprising, since it is well known that the assignment problem has a "unimodular" matrix and therefore the solutions for the x_{ij} are guaranteed to be 0-1 if we solve the problem as an LP.

It should be noted that a number of other modelling systems are available, including the AMPL system. AMPL is very similar in nature to GAMS. There are also chemical engineering modelling systems such as ASCEND and SPEED-UP which include capabilities for steady-state and dynamic simulation and nonlinear programming. SPEED-UP is distributed by Aspen Technology, while ASCEND can be obtained from Carnegie Mellon University. Both systems have capabilities for nonlinear programming.

Textbooks and Case Study

The two major textbooks available for teaching optimization in chemical engineering are:

Reklaitis, G.V., A. Ravindran and K. M. Ragsdell, *Engineering Optimization*, John Wiley (1983).

Edgar, T.F. and D.M. Himmelblau, *Optimization of Chemical Processes*, McGraw Hill (1988).

Both textbooks can be used at undergraduate and graduate levels. The text by Reklaitis et al. is more general and somewhat easier to follow. The book by Edgar and Himmelblau is more up to date and contains many examples relevant to chemical engineering. We have found that students like both textbooks, and that many of the exercises in them can be solved with LINDO, GINO or GAMS. Additional references appear in the bibliography.

In order to complement the above textbooks and to reinforce skills in problem formulation, CACHE has produced the case study:

> Vol. 6: *Chemical Engineering Optimization Models with GAMS*, CACHE Design Case Study.

The case study is based on the GAMS modelling system and covers applications at various levels of complexity in the following areas:

> Planning and scheduling of batch and continuous processes
>
> Chemical and phase equilibrium
>
> Design of heat exchanger networks, distillation columns, batch processes
>
> Synthesis of reaction paths, heat exchanger networks, distillation sequences
>
> Optimization of dynamic and distributed parameter models

This case study describes in some detail the formulation and solution of a total of 21 optimization problems that correspond to LP, MILP, NLP and MINLP models in the areas cited above. A special student version of GAMS for PC's is included that can handle problems with up to 1000 non-zero entries (up to 200 nonlinear) in the Jacobian and up to 20 discrete variables. The codes BDMLP, ZOOM, MINOS and DICOPT++ are included, as well as the *GAMS User's Guide*. Extensively documented GAMS input files are provided for each of these problems. Some of these input files are rather general, almost like self contained programs where problems can be specified with different data and dimensionality. This case study has been prepared by faculty and students at Carnegie Mellon University, Northwestern University and Princeton University. GAMS Development Corporation, the Licensing Technology Office at Stanford University, XMP Optimization Software and the Engineering Design Research Center at Carnegie Mellon have donated the computer software for this case study.

The case study is organized as follows. First a brief tutorial introduction to GAMS is given to illustrate at the simplest level some of its capabilities, and how to use the program. Also, some useful hints are included in this section. Next, the 21 problems listed in Table 1 are presented, with emphasis on the formulation of the problems. The GAMS input file for each problem is included (output listing files have been omitted except for the first problem because of space limitations), along with a discussion of results.

The GAMS input files of these problems are included in the diskette provided for this case study. A number of suggested exercises have also been included in the description of each problem. At the simplest level, they involve solving the same problem for different data; at the more complex level they involve modifying the formulations or explaining certain features in the model or algorithm. Several appendices are also included. The first one is a guide for the use of the MINLP optimization code DICOPT++, which is a non-standard option in the commercial version of GAMS. The second appendix shows how GAMS can be used to program an algorithm such as the Generalized Benders Decomposition. Finally, an appendix is included that provides some background on how to convert optimization problems with differential equations into nonlinear programming problems.

Table 1. Chemical Engineering Optimization Models with GAMS

1. LP model for refinery scheduling
2. LP model for production planning in multipurpose batch plants
3. LP transshipment model for predicting minimum utility consumption with constraints
4. LP/MILP model for reaction path synthesis
5. MILP model for the design of a chemical complex
6. MILP model for the scheduling of a multiproduct batch plant
7. MILP multiperiod model for the planning of chemical complexes
8. NLP model for power generation via fuel oil
9. NLP model for the optimization of an alkylation plant
10. NLP model for chemical equilibrium via minimization of Gibbs free energy
11. NLP model for phase and chemical equilibria via minimization of the Gibbs free energy
12. NLP model for global optimum search in the synthesis of heat exchanger networks
13. MINLP model for the selection of processes
14. MINLP model for the optimal design of multiproduct batch plants
15. MINLP model for simultaneous optimization in heat exchanger network synthesis
16. MINLP model for the synthesis of heat integrated distillation sequences
17. MINLP model for the synthesis of non-sharp distillation sequences
18. MINLP model for the optimal selection of feed tray in distillation columns
19. NLP differential/algebraic model for minimum time for a car to cover a fixed distance
20. NLP differential/algebraic model for optimal mixing of catalyst for packed bed reactor
21. NLP differential/algebraic model for parameter estimation in a batch reactor

Conclusions

Chemical engineering is a discipline in which many analysis, design and operations problems can be formulated as optimization models. Current courses and textbooks in chemical engineering optimization emphasize mostly theory and methods, and are restricted to rather small problems and applications. Although it is of course important to cover basic methods and the theory of optimization, it is also important to account for a number of new developments in this area. One of the most important developments is that new modelling systems such as LINDO,

GINO and GAMS offer the possibility to quickly model and solve a variety of optimization problems. Students can concentrate mostly on problem formulation, without having to spend too much effort on learning to use the software.

A second important development in optimization is that new and improved algorithms offer the possibility of solving much larger problems that could be handled previously. Linear programming problems involving thousands of constraints and thousands of variables can be readily solved. Similar trends are taking place for solution of mixed-integer linear programming problems. Another development is in the area of nonlinear programming, where currently one can solve problems involving several hundreds of equations and variables. Finally, important developments have also taken place in the solution of differential-algebraic systems and in the solution of mixed-integer nonlinear programming problems that, until recently, have received very little attention. There is a clear trend towards greatly improved capabilities for solving optimization models.

Given all of the above developments, there is an important educational need to reinforce the modelling skills of our students, to make use of both modern modelling systems and modern optimization algorithms, with which one can solve larger and more complex engineering problems. CACHE has produced a comprehensive case study to provide a set of chemical engineering problems to supplement optimization courses at both the undergraduate and graduate level. This case study is, of course, not meant to replace textbooks, but rather to provide additional material that will be useful for both instructors and students.

References

Bazaraa, M.S. and C.M. Shetty (1979). *Nonlinear Programming*, John Wiley.

Brooke, A., D. Kendrick and A. Meeraus (1988). *GAMS-A Users Guide*, Scientific Press.

Hillier, F.S. and G.J. Lieberman (1986). *Introduction to Operations Research*, Holden Day.

Liebman, J., L. Lasdon, L. Schrage and A. Warren (1986). *Modelling and Optimization with GINO*, Scientific Press.

Minoux, M. (1986). *Mathematical Programming: Theory and Algorithms*, John Wiley.

Nemhauser, G.L., A.H.G. Rinnoy Kan and M.J. Todd (eds). (1989). *Optimization, Handbook in Operations Research and Management Science*, Vol. 1, North Holland.

Schrage, L. (1984). *Linear Integer and Quadratic Programming with LINDO*, Scientific Press.

Williams, H.P. (1978). *Model Building in Mathematical Programming*, Wiley Interscience.

DESIGN CASE STUDIES

Ignacio Grossmann
Carnegie Mellon University
Pittsburgh, PA 15213

Manfred Morari
ETH Zentrum
Zurich CH-8092, Switzerland

Abstract

We discuss the role that design case studies play in undergraduate design courses and present a general overview of project assignment in design courses, and the evolution and trends of design case studies. A description of the CACHE design case studies is also given, followed by future trends.

Introduction

The design course is one of the more difficult courses to teach at the undergraduate level. A major problem has been the lack of design textbooks. From 1990 through 1995, only one new major design text, *Conceptual Design of Chemical Processes* by Douglas (1988), was published. This has meant that most faculty have had to put together course notes from a variety of different sources to keep up with new developments, design methodologies and computer tools. Another major problem in teaching design has been the lack of case studies that faculty can use as design projects. This means that faculty commonly spend a considerable amount of time to either get problems from industry or from the literature. CACHE, through its Process Design Case Studies Task Force, has tried to aid faculty with this problem.

The mission of the Process Design Case Studies Task Force is to promote the development of case studies for process design education to help instructors conduct design courses and provide examples for students. It is the chief objective of the CACHE Case Studies to demonstrate and elucidate the thought and decision process that is used in design, and to show how it can be supported by the use of computer tools for simulation, optimization and synthesis. Projects for the CACHE case studies have ranged from traditional to non-traditional processes, from grassroots to retrofit designs, and from large to small-scale problems. Task force activities include seeking and evaluating proposals for the development of case studies, reviewing case studies that are under development, and assisting in their final preparation, promoting recently developed case studies, acquiring design problem statements from industry, and organizing sessions and workshops at meetings dealing with process design education.

We give first a brief account of the common difficulties encountered in assigning and managing projects in the undergraduate design course, and then give an overview of the availability and evolution of design case studies. We then briefly describe the case studies that have been developed by CACHE. Finally, we discuss future plans and directions.

The Design Project

Teaching the design course is certainly a major challenge. In addition to covering lecture material, the instructor has to generate at least one design project. That generally means that the time spent in the course by both instructors and students is quite large, probably well beyond the credits earned. Furthermore, when you add the fact that difficulties may arise among the students in their groups, that they are not used to making decisions with incomplete information and that the grading of the project is somewhat subjective you have the perfect recipe for getting poor teaching ratings from students, which makes the teaching of the design course even more frustrating. While there are no magic recipes, one way of improving the teaching of the design course is by carefully selecting the design project and managing it in an efficient and meaningful manner for the students. Below we offer some general guidelines which clearly point to the usefulness of design case studies.

Firstly, there is the question as to whether it is better to assign one single problem to the whole class or a different one to each group of students. It may be a matter of personal preference, but we would recommend as a general rule to assign only one project. Unless the projects are relatively well defined or not very extensive, the work involved in gathering preliminary information, making sure the computer tools are applicable to the problem, preparing the teaching assistants, and keeping track of several projects simultaneously, is so great that it can become a very frustrating experience to assign different projects to different groups. Furthermore, even if only one project is given but it is sufficiently open ended, it will have multiple solutions, making it an ideal candidate for a design project. It is very interesting to see at the end how different alternatives are synthesized by the students.

The second major decision is whether the project should be a large design problem (e.g., design a complete flowsheet) or whether it should consist of solving perhaps two or more smaller problems. In the past, the practice has tended towards the first option. It has the advantage that students get to see what it takes to do a complete design. The drawback is that it exposes the students to only one problem which is what the second option seeks to avoid, as one can consider quite different applications with multiple projects. We personally feel that both options can be good experiences for students. The choice should be what the instructor believes he/she can do best.

The third major decision is how to integrate the design groups. In our experience what seems to work best is to let the students select their own partners within a specified deadline. Those who do not meet the deadline should then be assigned by the instructor. One can argue that this scheme has the disadvantage of creating unbalanced groups; presumably, some groups will be very strong, others will be weak. While this is generally true as far as the GPA is concerned, it is certainly not uncommon in design projects for some academically good students to do rather poorly and for some of the weaker students to do much better than expected. Furthermore, with our assignment scheme students feel more at ease because they make their own

decisions about whom they will work with. Otherwise a common problem is that students can blame the instructor for making them work with students they do not get along with.

The fourth major decision is how to organize and manage the project. Here perhaps the most important rule is that a fair amount of work must be completed before the project starts. In particular, it is important not only to have a problem statement, but to have some rough preliminary calculations to avoid unpleasant surprises (e.g., no matter what design alternative is selected, it always leads to an economic loss). One also should check that the library has the literature that students are likely to need and whether it is possible to use the process simulator or other computer tools for the process. It is also important to involve teaching assistants or other faculty at this stage to make sure they get familiar with the project so that they can supervise the design groups more effectively. Clearly, if the instructor has access to a design case study, the task of performing the preliminary work is facilitated greatly. Another aspect that must be considered is very clear specification deadlines for students to submit their memos, progress reports, and final reports, right from the beginning of the course.

Finally, the last major decision is how to grade the projects. It is important that students recognize that a project is a team effort. They should also recognize that good performance is rewarded and that poor performance is unacceptable. One particular scheme that we have used is to assign 50% of the grade to the group effort (e.g., one common grade for the memos and final design reports). The remaining 50% of the grade is individual. It is a *subjective* grade (students are told about this) that is assigned by the project supervisor. The specific scheme that we have used in the past is to assign individual grades so that the average comes out to be the same as the group grade. Aside from using the input by the supervisor of that group, we also use evaluations by the students themselves. At the time each memo or report is submitted, we provide confidential evaluation forms in which students indicate the percent contribution of each group member and any special comments. In our experience, about 80% of the groups indicate the same percentage contribution for each student.

As seen above, assigning a project in a design course is a nontrivial matter, that can be facilitated with design case studies, discussed next.

Overview and Evolution of Design Case Studies

One of the first major efforts in generating design case studies culminated in the series of problems produced by Washington University in St. Louis, MO. These case studies were developed in the 1970s and tended to address interesting, but generally well defined, problems. In a similar vein, AIChE has been producing student contest problems annually which, while often challenging, have been somewhat restrictive in scope.

Recently there have been increasing demands for assigning new and different types of design projects. Firstly, there has been the trend of requiring that the design problems be open ended and ill-defined to bring them closer to “real world problems” in which students have to work in the form of teams, rather than individually. Secondly, given research advances in the area of process synthesis, there has been an increasing emphasis on the application of systematic techniques at the early stages of design, in which decisions normally have a large economic impact. In particular, greater emphasis is being placed in the generation and screening of alter-

natives (e.g., hierarchical decomposition) and in the efficient recovery of energy (i.e., pinch analysis). Thirdly, tools for process simulation have become commonplace, especially given the significant advances in computer performance and the introduction of graphical user interfaces that greatly simplify the use of these complex programs. It has been the changing nature of these three developments, and the need to provide a service to faculty teaching design, that have motivated the work of the Process Design Case Studies Task Force over the last ten years.

The first five volumes of the CACHE Design Case Studies are shown in Table 1.

Table 1. CACHE Design Case Studies.

Volume 1	Separation System for Recovery of Ethylene and Light Products from a Naphtha Pyrolysis Gas Steam
Volume 2	Design of an Ammonia Synthesis Plant
Volume 3	Design of an Ethanol Dehydration Plant
Volume 4	Alternative Fermentation Processes for Ethanol Production and Economic Analysis
Volume 5	Retrofit of a Heat Exchanger Network and Design of a Multi-product Batch Plant

As will become apparent in the next section describing each of these volumes, the evolution of the CACHE case studies has proceeded as follows. The first three volumes were developed in the early 1980s and deal with the design and economic evaluation of large scale plants for manufacturing commodity chemicals (ethylene, ammonia, acetaldehyde). These design problems are fairly large in scope, and typically require groups of 4 to 6 students, although it is possible for an instructor to scale back the extent of these problems. Another common feature in these three case studies is the application of heuristics for most design decisions and of the pinch analysis for heat integration. Also, these three case studies illustrate extensive use of steady-state process simulators.

The fourth volume deals with the design of a fermentation process, to reflect the increasing interest in bioprocesses in the mid 1980s. The scope of the problem is similar to the first three, but it involves modification of a conventional simulator for handling nonstandard equipment and chemical compounds. Also, as with the first three volumes, special reactor models have to be developed.

The fifth volume represents a significant departure from the first four, in that it deals with two smaller problems that can be completed in 2-3 weeks and handled by groups of 2-3 students: retrofit of a heat exchanger network and design of a multiproduct batch plant. The intent here is to acquaint the student with some new trends that started to emerge in the late 1980s; the chemical industry was not building new large-scale plants, but retrofitting old ones, and it was concentrating on designs for manufacturing low volume specialty chemicals. Another interesting aspect of the fifth volume is that it assumes that the students will be working for a consulting firm, while the first four assume that the students will be working in divisions of

large corporations. Although unintended at the time this volume was produced, it seems to reflect the emerging working environment for engineers in the 1990s.

One important aspect in the five volumes of the CACHE Design Case Studies is that they contain special material for the instructor to help him/her to organize the course. Also, since the case studies are generally well written and presented, they can serve as examples for students on how to write and communicate effectively.

Summaries of Case Studies

We include here brief descriptions of the first five volumes of the CACHE Design Case Studies; each of the case study problems originated in actual design courses and have been used extensively in the U.S. and overseas.

Volume 1. Separation System for Recovery of Ethylene and Light Products from a Naphtha Pyrolysis Gas Stream

The objective of this case study is the design of a separation train for recovery of seven product fractions from steam cracked naphtha involving about two dozen components. All stages of the design procedure, from preliminary calculations to detailed flowsheet calculations, are described. Emphasis is placed on the following steps: preliminary synthesis of the separation sequence, optimization of column pressure levels for energy cost minimization, and synthesis of a heat exchanger network. Distillation is used for all separation operations except for methane/hydrogen, for which a membrane separator is proposed. The distillation columns were designed using DESPAC, an interactive program developed at Carnegie Mellon University. Any other standard program could be used for this purpose.

Depending on the required detail and the availability of suitable distillation column design software, the case study is suitable as a one-term assignment for either a single student or a group of students. The published case study is based on the work of five students over a ten week period. The problem was posed by Dan Maisel from Exxon and the case study was prepared by Michael Lenncoff under the supervision of Ignacio Grossmann and Gary Blau of Carnegie Mellon University.

Volume 2. Design of an Ammonia Synthesis Plant

The objective of this case study is the design of an ammonia synthesis plant that uses hydrogen and nitrogen feedstocks from a coal gasification plant. All stages of the design procedure, from preliminary calculations to detailed flowsheet calculations, are described. Emphasis is placed on the following steps: screening of key flowsheet decisions (pressure of synthesis loop, ammonia recovery, synthesis of gas recycle, hydrogen recovery from purge stream), selection of reactor configuration, cost minimization, and synthesis of the heat exchanger network. The proposed design incorporates a medium-pressure synthesis loop with water absorption/distillation for ammonia recovery, and with membrane separation for hydrogen recovery. The process was designed with the simulator PROCESS from Simulation Sciences, and the ammonia reactor was designed with the special-purpose package QBED. A listing of this program is included in the case study.

Depending on the required detail and the availability of process simulation software, the case study is suitable as a one-term assignment for either a single student or a group of three students, while the final design report is based on the work of a group of five students. The problem statement was supplied by Philip A. Ruziska from Exxon Chemicals, and the case study was prepared by Stacy Bike under the supervision of Ignacio Grossmann from Carnegie Mellon University.

Volume 3. Preliminary Design of an Ethanol Dehydrogenation Plant

The objective of this case study is the preliminary design of an acetaldehyde synthesis process by ethanol dehydrogenation. The project covered all stages of the design procedure starting from consideration of qualitative aspects of the flowsheet and preliminary calculations to detailed process simulations and final economic evaluations. In this study, emphasis is placed on synthesizing a workable flowsheet and justifying its configuration, simulating and evaluating the design using a commercial process simulator, and deriving a heat recovery network for the final process. The main reaction in this process is the endothermic dehydrogenation of ethanol to acetaldehyde. However, under the specified reactor conditions, a number of byproducts are produced and their presence determines a number of interesting alternatives for separation.

Once these alternatives have been screened and a workable flowsheet has been synthesized, the study centers on the simulation of this flowsheet using PROCESS from Simsci, Inc. Here, some of the features, advantages, and limitations of this simulator are presented. Finally, the study concludes with a complementary presentation of this process simulated with the CACHE version of FLOWTRAN. While the aim of this study is not to provide a detailed comparison between PROCESS and FLOWTRAN, a useful description of the relative merits of these simulators can be readily observed.

This project is suitable for a one-term project by a five or six person team of senior design students. The results of two such teams are given in this study. This problem was posed by the Union Carbide Corporation and the case study was prepared under the supervision of L.T. Biegler of Carnegie Mellon University and the late R. R. Hughes of the University of Wisconsin.

Volume 4. Alternative Fermentation Processes for Ethanol Production

The objective of this case study is the preliminary design and economic evaluation of a fermentation process for the production of ethanol from a molasses feedstock. The intent is to expose the student to some non-traditional chemical engineering processes and to the expanding field of biotechnology. The scope of the study is such that groups of 2-3 students should be able to complete the design in about 30 days. The major focus of this design is the creation and rational development of a suitable process flowsheet, simulation of the flowsheet by the simulator FLOWTRAN, and economic evaluation and cost minimization of the final process.

The problem begins with the specification of the plant operating requirements. The type of fermentor to be used as well as plant operating conditions are left open. Suggested fermentors include batch, CSTR, and CSTR with cell recycle, as well as a novel extractive fermentor based on the use of hollow fiber membranes (HFEF). The choice of the fermentor will affect the nature of the flowsheet and lead to several design alternatives that the students will have to screen before arriving at a workable flowsheet that is ready for simulation. This case study in-

cludes a floppy disk with input files for the simulator FLOWTRAN and a program written in BASIC to evaluate the performance of CSTR fermentors.The problem statement was posed by Steven LeBlanc and Ronald L. Fournier and prepared under their supervision by the student Samer Naser at the University of Toledo.

Volume 5. Retrofit of a Heat Exchanger Network and Design of a Multiproduct Batch Plant

This volume contains two short design projects that can be developed by groups of 2-3 students in about two weeks. As opposed to the large projects that are commonly used in a design course, the objective of the case study is to expose students to a greater variety of problems and which are of current industrial significance.

The first problem deals with the retrofit of a heat exchanger network consisting of 8 exchangers with 5 hot and 3 cold processing streams as well as steam and cooling water. The layout of the network and areas of the exchangers are also given. The objective is to determine a retrofit design that can reduce the energy consumption within specified limits for the capital investment and payout times. This problem requires examination of alternatives for the level of energy recovery, matching of streams, addition of area, and removal or reassignment of existing exchangers and piping. This problem can be used to illustrate basic concepts of heat integration, as well as the application of computer software such as Target II, THEN, MAGNETS and RESHEX.

The second problem deals with the design of a batch processing plant that has to manufacture 4 different products, all of which require 5 similar processing steps (reaction, product recovery, purification, crystallization and centrifuge). An important aspect of this problem is that the production schedule and inventory must be anticipated at the design stage. Furthermore, this problem also requires analyzing alternatives for merging processing tasks into single units, and using parallel units with and without intermediate storage. The use of Gantt charts is emphasized to examine some of these alternatives. The case study also includes two sets of homework problems with solutions that can be used to provide the basic background for the two problems. This case study has been prepared by the students Richard Koehler and Brenda Raich at Carnegie Mellon University under the supervision of Ignacio Grossmann, who developed the problem statements and educational material.

Concluding Remarks and Future Directions

Though no two design problems are alike, there is a general logical sequence of basic steps which lead to a good design. It is the chief objective of the CACHE Case Studies to demonstrate and elucidate this thought and decision process. The CACHE Case Studies are different from final student or industrial project reports in that they not only present one final solution, but show the whole solution procedure leading from the problem statement to the final solution(s) in an organized manner. This chapter has discussed general issues involved in projects for the undergraduate design course. The role of design case studies to facilitate this task has been discussed. Finally the first five volumes of the CACHE Process Design Case Studies have been described.

It should be noted that the sixth volume of this series, *Chemical Engineering Optimization*

Models with GAMS, is devoted to optimization. This case study is described in the preceding chapter and would normally be used in an optimization course, although parts of it might also be incorporated into a design course, if optimization is taught as part of it. Also, a new case study on process operations (data reconciliation in a refinery) is under development at McMaster University and will be available soon.

As for future trends, we believe that the design course will continue to be the major capstone course in chemical engineering. It is the only course in the curriculum in which the students put together and apply all the material they have learned in chemical engineering. It is a course in which students have to make decisions, often with limited or incomplete information. It is also a course in which students have to learn how to work and get along with their group members. Finally, it is a course in which students learn the importance of written communication as a complement to the technical work.

It should also be noted that the teaching of the design course will be facilitated in the future with at least two new design textbooks. *Chemical Process Design* by Robin Smith (1996) has just been published, while *Systematic Methods for Chemical Process Design* by L. Biegler, I. Grossmann, and A. Westerberg is due to appear in 1997.

The design project, however, will continue to play a central role in the design course. In order for the project to fulfill a relevant and timely role, it is important that future case studies address nonconventional processes (e.g., electronics) and pressing industrial problems (e.g., environmental). It is also important that the projects be directed towards operational issues. The case studies should incorporate the use of new computer tools and environments, and reflect new trends in industrial practice. We hope that CACHE can continue to promote the creation and distribution of such case studies, and encourage faculty who can develop good design problems to participate with CACHE in this endeavor.

PROCESS CONTROL

Yaman Arkun
Georgia Institute of Technology
Atlanta, GA 30332

Carlos E. Garcia
Shell Oil Company
Deer Park, TX 77536

Abstract

This document summarizes CACHE's involvement in process control education during the last 25 years. A historical review of past activities is given; current trends are pointed out and a futuristic perspective on chemical process control education is provided.

Introduction

We review and put into perspective CACHE's involvement in process control education over the last 25 years, hoping that this information will prove useful for chemical engineering educators. As in essentially all of its activities, CACHE's role here has been primarily to facilitate the introduction of computer aids and define new educational directions in process control. In doing so, CACHE has tried to delineate innovative ways to use computer tools to develop creative teaching media which will help to improve the learning skills of our students.

Process control in the chemical engineering curriculum occupies a unique place. Firstly, its boundaries are not as rigid or well-defined as in some other courses. It requires a working knowledge and synthesis of many diverse and interdisciplinary concepts ranging from basic principles of modeling to controller design. Secondly, it studies techniques and computer algorithms which are used in the chemical process industry in connection with real-time monitoring and digital control. Therefore, in addition to the basics, some exposure to computer implementation is also important. Considering the computational challenges faced during modeling, dynamic analysis and controller synthesis, and the technological advances made in industrial computer control practice, it is not surprising that process control education has witnessed increasing use of computer hardware and software over the years. This transformation was accompanied by interesting debates on the appropriate course coverage to bridge the gap between academic training and industrial practice (Edgar, 1990). In what follows, we review CACHE's role in this evolution by describing its major activities and the educational materials it has introduced for control courses. Finally we will express our views on the current and future directions of process control education.

CACHE's involvement in process control education dates back to the early 1970s. In these earlier days of its establishment, with the help of many faculty from different universities, CACHE developed 150 computer programs and modular instruction materials (Edgar et al., 1988) for core chemical engineering courses, including process control. These FORTRAN programs included the following control modules: roots ofthe characteristic equation, bode plots of transfer functions, control of a pure timedelay, simulation of tank level control, and frequency response data via pulse testing. They became quite popular and were reprinted three times. These programs relieved students from repetitive and tedious calculations and allowed them to spend more time in creative problem solving. It was also the first time that interactive computer graphics had been introduced into the chemical engineering curriculum. At the time (1976), the first author was in graduate school performing TA duties in an undergraduate process control course. He remembers vividly the excitement caused by the introduction of root locus and bode plots. CACHE was also involved in preparation of instructional process control modules published by AIChE in five volumes (AIChEMI, 1981-1986), and which are still available.

Recognizing the importance of interactive graphics, CACHE formed a task force on computer graphics in the late seventies. Its main goal was to develop an awareness of the power and pedagogical value of interactive graphics in chemical engineering education. Its activities included publication of tutorials and position papers providing information on interactive graphics devices, and recommended hardware configurations and potential applications (Carnahan et al., 1978; Reklaitis et al., 1983). From these studies, process control emerged as a prime candidate to benefit from the use of interactive graphics tools. In particular, graphical tools for controller tuning, such as Nyquist plots, became more widely used as teaching tools. Several departments began development of their own dynamic simulators and design tools with graphical interfaces.

A CACHE survey carried out in the summer of 1981 revealed that while access to graphics hardware in chemical engineering departments was growing, educational use was still quite limited by the availability of general purpose software, and hardware consisted mostly of older monochrome Textronix devices. In retrospect, the picture has changed drastically in the last decade. Starting with Program CC (Thompson, 1985) and continuing with MATLAB, general purpose commercial packages have taken the place of much in-house control software in recent years. At the same time, rapidly growing computer technology has allowed use of interactive graphics on many different hardware platforms, including PCs, Macs and UNIX workstations, all at affordable prices. Today, commercial packages such as MATLAB are used in quite a few chemical engineering departments (Kantor, 1992). More will be said about the current status of these tools in process control education later on.

One of the most significant commitments of CACHE to process control education has been in the area of real-time computing. Motivated by the lack of educational material in this area in the 1970s, the Real-Time Computing Task Force published eight introductory monographs (Mellichamp, 1977) that covered basics of real-time programming and architectures, prototype laboratory experiments, computer interfaces and application software, and data acquisition and control algorithms. These monographs, which later became a textbook (Mellichamp, 1983), were in high demand for teaching real-time systems and are still being used in some digital process control courses. In parallel with CACHE activities, integration of real-time computing into process control teaching had already started in the late seventies in quite

a few chemical engineering departments (Morari and Ray, 1979).

Faced with the pervasive use of personal computers in the1980s, CACHE, following its traditional interest in real-time computing, took a new initiative, "microcomputer applications in the laboratory". The mission was to assist chemical engineering departments in developing undergraduate laboratory experiments involving on-line computer applications. The motivation was to give a picture of the wide array of classical and modern engineering concepts that have been successfully built into computer-aided laboratory experiments, of the many very different hardware and software approaches, and of the pedagogical objectives to be fulfilled when the computer was used for control purposes. The specific CACHE contributions included a *Survey* of U.S. and foreign chemical engineering departments (Cinar and Mellichamp, 1986), an *Anthology* (Cinar and Mellichamp, 1988), and a publication describing Real-Time Task Force activities (Arkun, et al., 1989). The *Anthology* contains descriptions of twenty one experiments covering the areas of thermodynamics, fluid dynamics, heat and mass transfer, chemical and biochemical reactors, process dynamics, and control. In process control education this document should help, especially faculty who are developing on-line computer control applications for the first time. There is no doubt that teaching process control with this type of laboratory instruction is very desirable, as the students are introduced to real problems, and acquire valuable hands-on experience with control hardware and different data acquisition and control practices.

In the 1990s, CACHE has been involved with two major products, the Model Predictive Control Toolbox (CACHE-Tools) (Morari, 1993) and PICLES (Cooper, 1993). While CACHE-Tools have been developed as a CACHE task force activity, PICLES was created at the University of Connecticut by Douglas Cooper with CACHE assistance in testing and disseminating the program. PICLES (Process Identification and Control Laboratory Experiment Simulator) is a training simulator that provides access to several simulated processes with real world behaviors; processes include gravity drained tanks, a heat exchanger, a pumped tank and a binary distillation column. It also includes "mystery processes", with models hidden from the students, in which input and output data are displayed for subsequent identification and control studies. In all these processes, PICLES demonstrates the significance of nonlinearites, time delays, noisy measurements, inverse and integrating open-loop responses, and multivariable interactions. The available controllers include different PID algorithms, Smith predictor, feed forward control and a decoupler. PICLES provides a color graphic display of process equipment with some animation, historical data of inputs and outputs, and controller templates (see Fig. 1). The program runs on IBM-compatible personal computers. It can be incorporated easily into existing undergraduate process control courses either by assigning homework around the processes it provides or by treating it as a simulated experiment in the control laboratory.

The commitment to bring new and proven industrial control practices into academic training is best reflected in the recently introduced software, CACHE-Tools. This is a collection of MATLAB functions developed for the analysis and design of model predictive control (MPC) systems. MATLAB runs on almost all personal computers and engineering workstations and is available at a significant educational discount. MPC was conceived in the 1970s primarily for industry. It can be now safely be argued that it is the most widely used multivariable control algorithm in the chemical process industries. MPC is a novel technique for feedback control that makes explicit use of a process model. It uses the model to predict the effect of control

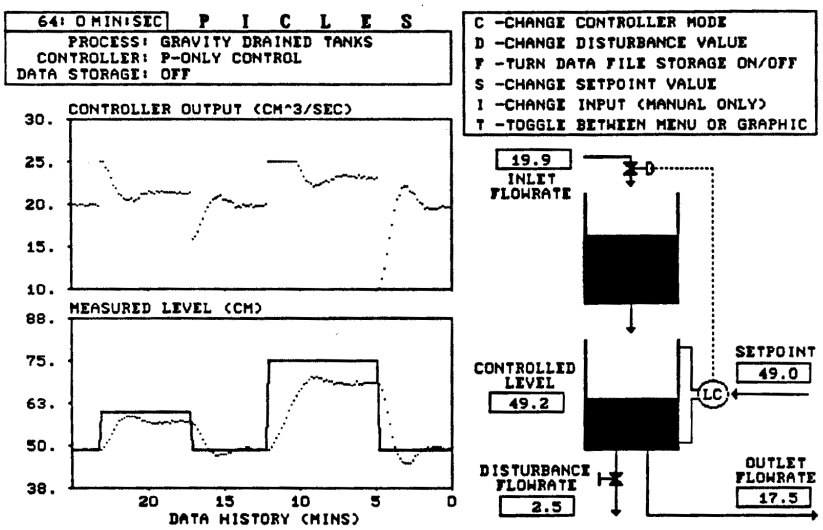


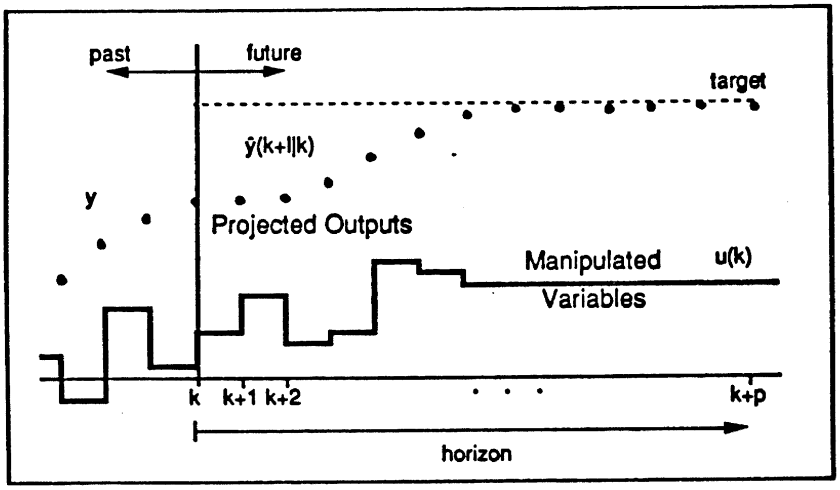
Figure 1. PICLES - Gravity-Drained Tanks under P-Only control shows increaing offset as set point moves further from design value.

actions and to estimate disturbances (see Fig. 2). The control actions are calculated from an on-line optimization. The fact that a process model is involved makes MPC quite useful for teaching, since the interesting interplay between dynamic modeling and control can be taught and explored in a design setting that has immediate industrial applicability. We can now appreciate better the significance of different types of process models for control purposes. Also, with MPC, real and difficult process problems such as multivariable interactions, delays, measurable disturbances, an, most importantly, constraints on both process inputs and outputs, can be handled directly.

CACHE-Tools is intended for both the classroom and the practicing engineer. It can help in communicating the concepts of MPC to students in an introductory course. At the same time it is sophisticated enough to use in higher level control courses and also to train practicing engineers. Recent undergraduate textbooks (Seborg, et al., 1989; Deshpande and Ash, 1988) include some introductory material on MPC. However, more tutorial teaching material is needed, and we expect the monograph (Morari, et al., 1994) will be valuable in this regard.

Finally, CACHE has been quite active in organizing process control conferences. CPC (Chemical Process Control) has become the premier international conference series bringing together participants from academia and industry. Although these conferences are primarily research oriented, they have impacted process control education by providing a forum for discussing industrial viewpoints and new educational paradigms (Downs and Doss, 1991).

CACHE's major activities in process control in the last 25 years is summarized in Fig. 3.



•Use model to predict the future changes of controlled variables.

•Correct the prediction using the disturbance effect estimate.

•Calculate manipulated variable moves to:

min[{prediction-setpoint}+{control effort}] subject to constraints on inputs

Figure 2. Model Predictive Control.

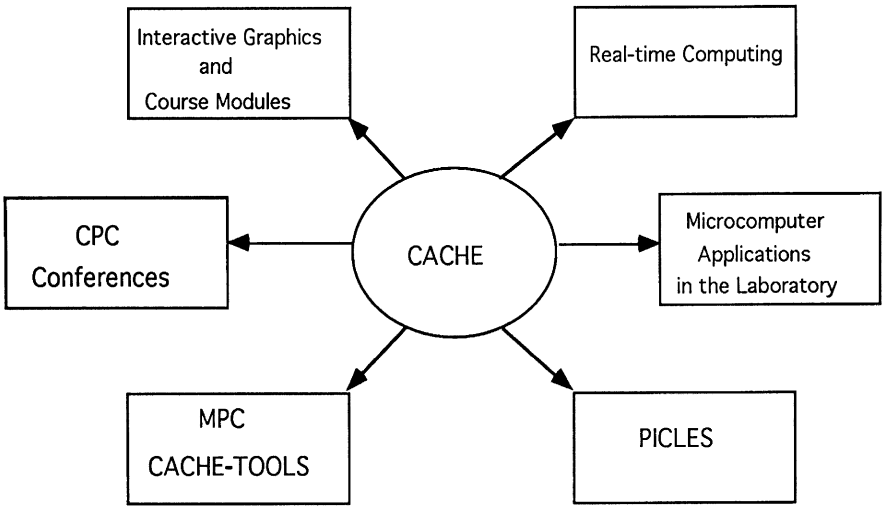


Figure 3. CACHE activities in process control.

Present Status and Future Trends

Today there seem to be two dominant trends in the development of computer aided educational materials. One is to create tool boxes around commercially available general purpose software. The other is the case study approach. For example, MPC CACHE-Tools has been designed around MATLAB, which is readily accessible and offers many useful capabilities, including symbolic manipulation, identification, control system analysis and design, digital signal processing, statistical analysis, and optimization. Any of these features can be used in connection with CACHE-Tools, if needed. In addition, a separate software package, SIMULINK, has added a nonlinear dynamic simulation and modeling environment to MATLAB's analysis environment. With the help of SIMULINK, different process case studies can be developed and interfaced with MPC or any other control algorithm. In fact, today quite a few departments are in the process of creating these types of case study modules. The first author has used SIMULINK control modules as part of a design project in an undergraduate control course in the last three or so years (Fig. 4). In the future, it would be valuable to document and distribute a library of such case studies to process control instructors. Finally, we should draw attention to the efforts of our colleagues devoted to process control education independent of CACHE activities. There have been numerous valuable contributions in the creation of educational materials, including software development, some of which is documented in Seborg, Edgar and Mellichamp (1989). Among these, in particular, CONSYD from Wisconsin, and UC-Online from Berkeley have contributed significantly to the teaching of process control.

Most of the changes that have occurred in process control education (and practice) in the last 25 years can be attributed to the introduction of faster and cheaper microcomputers. This has forced process control education to shift in emphasis toward "softer" chemical process operation improvement and away from the traditional "hard" instrumentation emphasis. Although theoretical course subjects have not changed in any significant way, we have seen more coverage of digital control with laboratory experiments, introduction of new computer algorithms like MPC, and some exposure to statistical process control. Basically, process control education has gone through a period of evaluation of new topics and teaching techniques in response to rapidly changing computer hardware and software developments, and new advances in control research and industrial practices.

We anticipate that the next twenty five years will be even more revolutionary. With increased implementation of advanced control systems and techniques in industry and with increased investment in sophisticated process control equipment, engineers need to be better educated in maximizing the use and value of control technology in manufacturing operations. This requires the plant engineer to have a thorough understanding of control technology and of the processes and the skills needed to successfully combine them to operate a plant profitably. Currently, this task is delegated to the "control engineer" or to vendors who, because of their lack of day to day process exposure, are limited in identifying opportunities. To achieve this, we see a long term challenge to shift the way engineers are educated away from steady-state design of unit operations and more into how plants are operated in real time. It is conceivable that every unit operations course could cover dynamic transient concepts and education on how these units are operated. The process control course will then become more broad and dedicated to teach all technology that drives those processes to satisfy operating objectives: control theory and algorithms, advisory and information systems, optimization, etc. Laboratory work

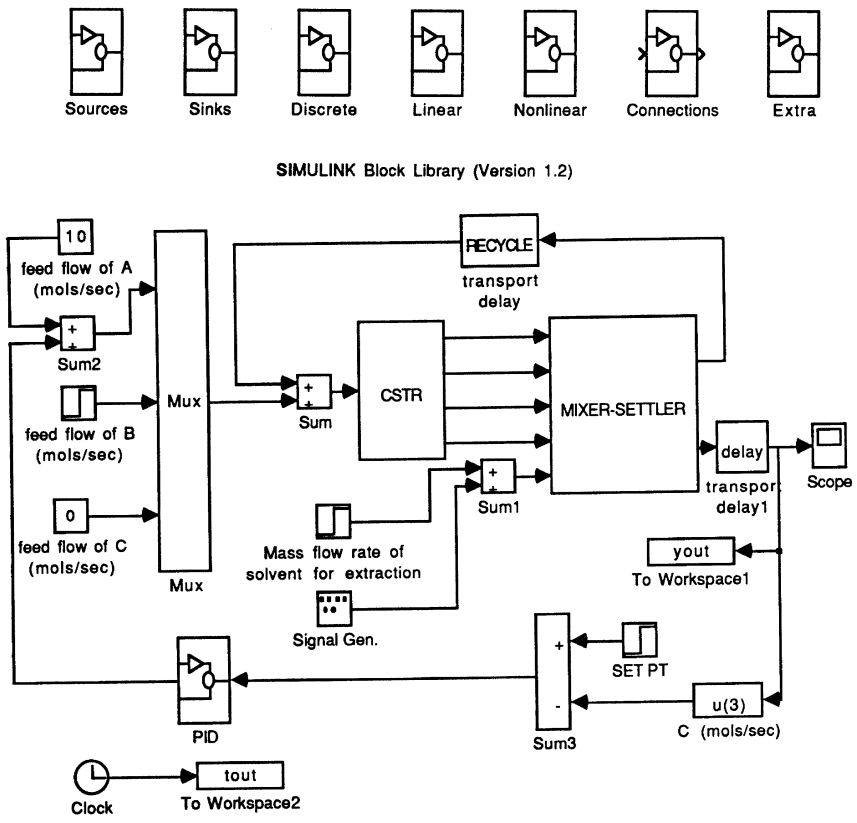


Figure 4. SIMULINK Module - Control of a reactor system with recycle.

will be used to emphasize the integration of these skills with process operation through control hardware and software.

Because of these trends, educational aids for control should help engineers learn the process dynamics as well as the control principles. Therefore, there will be a need for dynamic simulators with computer interfaces like those used in industrial operations. We also need educational aids that help in learning the mathematical concepts, the development of control strategies and control algorithm design, and the art of evaluating whether control systems are helping meet operating objectives. Therefore, topics such as process monitoring, statistical analysis and performance evaluation, model identification and parameter estimation, and large scale computing and optimization will need to be integrated with traditional process control education and covered in more depth in undergraduate courses. At the same time, in laboratories we expect to see more advanced instrumentation with friendly real-time interfaces, increased computational power and data analysis capabilities, advanced multivariable control algorithms, and less traditional experiments. Laboratory experience should prepare students for correctly using control skills on processes.

A most important area which will impact the way we teach process control in the future is dynamic modeling. Models are an excellent way to learn process dynamics and of course are the main requirement for control design and implementation. Systematic development and utilization of the appropriate models for different control purposes and analysis of effects of modeling inaccuracies on controller performance will find a permanent place in control education, as we learn more from research findings in these areas.

Also, as we broaden the impact of process control in operations, two challenging topics that have been haunting us for quite some time in education will increase in importance. These are integration of control with process design, and plant wide control (as opposed to unit operations control). Despite the need expressed by industry (Downs and Doss, 1991) and increased coverage in recent textbooks (Seborg, et al., 1989; Stephanopoulos, 1985), it is not clear how to teach these concepts and the demanding design tasks involved in the most efficient way. There must be more activity that sheds light in this direction, if process control education is to make a big impact on chemical engineering practice.

In summary, efficiencies in computing and the continued development of improved modeling and control tools better integrate this technology with plant operations and educational needs to reflect that trend. We see computer-based educational aids in control as being the key in teaching this fundamental technology to engineers, and expect CACHE to have a continued and major emphasis on process control education in the future.

References

AIChEMI Modular Instruction Series; published by AIChE: New York, 1981-86: Series A, Process Control; T. F. Edgar, Ed.

Arkun,Y., A. Cinar, and D. A. Mellichamp (1989). *Laboratory Applications of Microcomputers*. CACHE News, Spring.

Carnahan, B., R. S. H. Mah, and H. S. Fogler (1978). *Computer Graphics in Chemical Engineering Education.* CACHE: Cambridge, MA, June.

Cinar, A. and D. A. Mellichamp (1986). *Survey: Experiments Involving On-Line Computer Applications in Chemical Engineering Undergraduate Laboratories*. CACHE: Austin, TX, Sept.

Cinar, A. and D. A. Mellichamp (1988). *On-Line Computer Applications in the Undergraduate Chemical Engineering Laboratory: A CACHE Anthology*. CACHE: Austin, TX, Sept.

Computer Programs for Chemical Engineering Education. Sterling Swift Publishing: Austin, TX, (1972).

Cooper, D. J. (1993). PICLES: The Process Identification and Control Laboratory Experiment Simulator. CACHE News, Fall.

Deshpande, P. B., and R. H. Ash (1988). *Computer Process Control with Advanced Control Applications*, 2nd Ed., ISA.

Downs, J. J., and J. E. Doss (1991). The View from North American Industry, *CPC IV Proc.,* Y. Arkun and W.H. Ray, Eds., AIChE.

Edgar, T. F. (1990). Process Control Education in the Year 2000. *Chem. Eng. Edu*, Spring, 72-77.

Edgar, T. F., R. S. H. Mah, G. V. Reklaitis, and D. M. Himmelblau (1988). Computer Aids in Chemical Engineering Education. *CHEMTECH*, **18**, 277-283.

Kantor, J. C. (1992). Matrix Oriented Computation Using Matlab, CACHE News.

MATLAB, The MathWorks, Inc., South Natick, MA.

Mellichamp, D. A. (ed.) (1977). *CACHE Monograph Series in Real-Time Computing*. CACHE: Cambridge, MA.

Mellichamp, D. A. (ed.) (1983). *Real-Time Computing*. Van Nostrand-Reinhold: New York.

Morari, M. (1993). CACHE News.

Morari, M and W. H. Ray (1979). The Integration of Real-Time Computing into Process Control Technology, *Chem. End. Edu.*, 160-162.

Morari, M., C. E. Garcia, J. H. Lee, and D. M. Prett (1994). *Model Predictive Control*. Prentice Hall.

Reklaitis, G. V., R. S. H. Mah, and T. F. Edgar (1983). *Computer Graphics in ChE Curriculum*. CACHE: Salt Lake City.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp (1989). *Process Dynamics and Control*. John Wiley & Sons.

Stephanopoulos, G. (1985). *Chemical Process Control*. Prentice Hall.

Thompson, P. T. (1985). *Program CC*. System Technology, Inc., Hawthorne, CA.