MULTI-SOLVER MODELING FOR PROCESS SIMULATION AND OPTIMIZATION

L. T. Biegler and Dilek Alkaya
Carnegie Mellon University
Pittsburgh, PA 15213

Kenneth J. Anselmo
Air Products and Chemicals, Inc.
Allentown, PA 18195

Abstract

With the increasing size and complexity of process simulation and optimization problems, exploitation of the process model becomes increasingly important. While most researchers recognize the need to exploit the problem structure, for instance at the linear algebra level, this study explores the case when multiple model solvers are required for simulation and optimization of the overall process system. While this approach is standard for process flowsheeting, we need to consider how we can take advantage of sophisticated simultaneous solution and optimization strategies for large-scale optimization. Here we discuss both open form and closed form models, and demonstrate that both are needed for different types of problems. We then consider an approach where closed form or 'black box' models can be 'opened up' to achieve simultaneous optimization without disturbing the inherent structure of the model's solver. In addition, several applications, including process flowsheets, dynamic optimization, PDE models and process integration are highlighted. Finally, we close with some challenges and areas for future work for both modeling environments and optimization algorithms.

Keywords

Process optimization, SQP algorithms, open form (equation oriented) modeling, closed form modeling, tailored optimization, process integration.

1. Introduction

With increasing complexity and size of chemical process models, there remains a constant challenge to develop more efficient and robust algorithms for simulation and optimization. Here model developers and engineers are faced with conflicting objectives. First, there is a need to develop robust general purpose modeling platforms and numerical algorithms that allow the rapid development of large process models. This paradigm allows a straightforward and seamless extension to very efficient optimization algorithms as well. And modeling, simulation and optimization tools that are built on these concepts allow us to conduct large-scale optimization studies with the same ease as solving the process model itself. Excellent illustrations in this conference include the papers by Wright (1999), Floudas and Pardalos (1999), and Grossmann and Hooker (1999).

On the other hand, increasing complexity of these models requires us to consider the exploitation of specific model structure and the development of specialized solvers for different kinds of systems. This approach has its roots in the early history of process simulation. For instance, equilibrium stage models exploit the block tridiagonal structure of the MESH equations and related systems. Moreover, further solver decompositions like the inside-out algorithm must be applied in order to isolate and control highly nonlinear characteristics of some columns. Moreover, the exploitation of structure and specialized solvers is especially acute in the solution of PDE-based (Partial Differential Equation) models, where spatial and temporal discretizations create systems with millions of variables and equations. Here the 'one size fits all' solver paradigm breaks down. Excellent illustrations at this conference include the papers by Sinclair (1999) and Jensen (1999). However, for these systems, the extension of these models to optimization and integration with other process systems still remains a serious challenge.

This study attempts to bridge between highly developed process simulation and optimization tools and complex, diverse models that have yet to be considered in these environments. In the next section we begin by tracing the evolution of different process optimization strategies. In particular, we highlight the merits of simultaneous optimization strategies, both for efficiency and the ability to integrate different tasks and subsystems. Following this section, we highlight benefits and challenges required for optimization applied to process integration. Section 4 then discusses the current state of open form and closed form systems, i.e., models expressed in a declarative, equation oriented way, vs. models that are closely coupled and exploited by a special purpose solver. The latter case also includes the presence of legacy codes and also covers a wide variety of detailed models that are not widely considered in process flowsheeting or real-time optimization. As will be shown in this section, both open and closed form paradigms are essential for their respective model types. Section 5 develops a tailored optimization strategy that extends the benefits of simultaneous optimization to a wide variety of closed form models. Derived as a variation of the Successive Quadratic Programming (SQP) algorithm, this approach has been demonstrated on a number of different process systems. Further challenges for nonlinear programming algorithms are discussed in section 6 and section 7 briefly summarizes and concludes the paper.

2. A hierarchy of optimization approaches

In a recent lecture at the SIAM Optimization Conference, Betts (1999) provided a personal summary on the evolution of optimization applications in the aerospace industry. The main stages of this evolution can be classified through the use of *surrogate*, *black box* (with exact derivatives) and *simultaneous* optimization strategies applied to complex engineering models. In the *surrogate* approach, a simplified optimization problem is formed from easily obtained information from a detailed model. This problem is updated and solved repeatedly as the optimization proceeds and there is only a weak connection between the detailed model and the optimization process. In the *black box* approach, the detailed model is called repeatedly by the optimization algorithm in order to obtain accurate function (and gradient) information. Finally, the *simultaneous* optimization strategy leads to solution of the optimization problem and the complex model at the same time.

As one advances from the surrogate to the simultaneous strategies, it is clear that both the efficiency of the optimization approach and the scale of the application can be increased greatly. Also, the interactions between the detailed model and the optimization algorithm become much tighter, and considerably more effort is required to formulate the optimization problem. On the other hand, detailed models cannot all be adapted to optimization formulations in the same way. For some models, the application of simultaneous

strategies can be difficult and possibly counterproductive with current approaches. For instance, at Boeing Corporation, orbit trajectory optimization is a well established and highly efficient simultaneous optimization procedure (Betts and Frank, 1994), while more complex and detailed structural and fluid flow optimizations for aircraft design are currently handled by more primitive surrogate methods (Booker et al., 1998).

Similarly, it is instructive to consider the evolution of optimization methods in process engineering. The *surrogate approach* applies to any strategy that forms a simplified model of the system, which can be optimized directly. In this context we assume that this simplification can be obtained quickly and cheaply from the complex model (e.g., with perhaps a few function evaluations). For these approaches, one can cite pattern searches developed in the 60s (Hooke-Jeeves, Nelder-Mead, EVOP, etc.) and related stochastic strategies such as simulated annealing and genetic algorithms. Interestingly, many process synthesis approaches such as the use of residue curve maps, attainable regions and pinch curves for heat or mass integration can also be viewed as surrogate optimization methods. These methods are easy to apply and require little interaction between the rigorous, detailed process model and the optimization method. As a result, these methods are very popular for 'one-shot' optimization studies. On the other hand, their limitations arise in dealing with highly constrained problems and in large-scale and routine applications where efficiency is crucial.

The *black box* approach using gradient based methods is a popular procedure both for flowsheet optimization (e.g., ASPEN/Plus, ProSim, etc.) as well as the optimization of dynamic systems (gProms, ABACUSS). For efficient optimization strategies, a crucial component is the efficient and accurate calculation of derivative information. Nowadays, this task can be addressed through the application of efficient automatic differentiation (AD) and sensitivity strategies. For steady state models, AD approaches can be applied directly to the source code, so that exact derivatives with respect to any number of input variables are obtained with only a small multiple of the solution cost. AD (Griewank, 1989) has been applied to a wide variety of large-scale finite element and process models by replicating the calculation tree with corresponding derivative calculations. Nevertheless, this approach needs to be approached carefully to avoid differentiating through fixed point loops in internal calculations. Similarly, efficient sensitivity calculations have been adapted to parallel the solution of DAE systems (Feehery et al., 1998; Li et al., 1999). As a result, accurate gradients for Newton-type convergence and optimization can be obtained, often with only a small multiple of the solution cost.

Finally, *simultaneous methods* have seen widespread use in real-time optimization of petrochemical plants. Often used with 'open form' or equation based models, this strategy requires close collaboration of the modeler and the developers and users of the optimization strategy. Moreover, to obtain the performance benefits of this optimization approach, the development, analysis and implementation are much more difficult and time consuming than with the previous ones. Perhaps the greatest advantage of these approaches is their transparency to sophisticated general purpose numerical algorithms. Derivatives are calculated in an efficient and accurate manner and, in principle, the solver has full access to all variables, equations and derivative information.

Table 1: Simulation Time Equivalents on Evolution of Methods

Surrogate Models

•Friedman and Pinder (1972) 75-150
•Gaines and Gaddy (1976) 300

Black Box (with derivatives)

•Parker and Hughes (1981) 64
•Biegler and Hughes (1981) 13

Partially Simultaneous, Black Box

Biegler and Hughes (1982)
Chen and Stadtherr (1985)
Kaijaluoto (1984)
ASPEN+, PRO/II, HYSYS 10-30

Wolbert et al (1994) 3- 10

Simultaneous

•Locke and Westerberg (1983) < 5
•Stephenson and Shewchuk (1986) 2
•RTOPT, NOVA, etc. ~1

Table 1 summarizes the performance characteristics for flowsheet optimization using a few selected studies as benchmarks. Using a process flowsheet with about ten decision variables, we see requirements of hundreds (or even thousands) of simulation time equivalents for surrogate strategies. On the other hand, with simultaneous strategies, optimization requires virtually the same effort as the solution of the process model. Also, we observe reductions of over an order of magnitude when passing from one strategy to the next. Note that in the case of partially simultaneous, black box methods, where derivatives are calculated very efficiently (Wolbert et al., 1994), there is some overlap in performance with simultaneous strategies.

It is clear from this discussion that simultaneous strategies provide clear performance benefits. Moreover, these benefits translate into the ability to model and solve much larger optimization problems. As discussed in the next section, the ability to integrate multiple process subsystems and multiple design tasks (e.g., economics, operability, controllability, safety, energy recovery) leads to more important benefits than increase in performance alone.

3. Optimization - A Tool for Integrated Process Engineering

Over the past decade the integration of tools and process design environments has become a major activity in the process industries. The impacts of this integration on standardization of work processes and incorporating design and operation issues into the supply chain (Ramage, 1998; van Schijndel and Pistikopoulos, 1999) are widely recognized as key corporate activities. Marquardt and Nagl (1998) surveyed the development of standards for tool integration and classified these as: *Presentation integration* – integrated tool set with common look and feel presented to user; *Data integration* – sharing and managing relations among data objects; *Control integration* – notification and activation of tools using e.g., message passing and *Platform integration* – execution of integrated suite of tools on heterogeneous, distributed computer network. Examples of these standards include STEP and PDXI for data integration as well as CAPE-OPEN for data and control integration. Moreover, commercial examples of tool integration include VTPLAN (Bayer), Plantelligence (Aspen Tech) and SimSight/Simulation Manager (SimSci/Bayer). Academic projects in this area include efforts to support conceptual, design and front end engineering and are exemplified by the *n*-dim (Carnegie Mellon), KBDS/epee (Edinburgh), and the IMPROVE and CHEOPS (Aachen) systems.

More recently, Marquardt and coworkers (Backx et al., 1998; Helbig et al., 1998; Marquardt, 1999) demonstrated the importance of optimization tools within this integrated framework. Clearly the ability to model and optimize over entire systems and over multiple attributes leads to far superior solutions. Moreover, the integration of optimization formulations has been a fruitful activity in process systems engineering over the past decade. Studies include integration of batch process design and scheduling (Birewar and Grossmann, 1989; Voudouris and Grossmann, 1993), design under uncertainty, (Ierapetritou et al., 1996; Pistikopoulos, 1997), design and dynamic performance (Logsdon and Biegler, 1993), design, scheduling and dynamic performance (Bhatia and Biegler, 1996), scheduling and dynamic performance (Mujtaba and Macchietto, 1993), interactions of energy, separation and reactor subsystems (Balakrishna and Biegler, 1996; Duran and Grossmann, 1986), interactions of control and design (Luyben and Floudas, 1994; Morari and Perkins, 1994; Walsh and Perkins, 1994), process design and planning (Pinto and Grossmann, 1994; Sahinidis and Grossmann, 1991) and safety, design and performance (Abel et al., 1998).

To show the quantitative benefits of integration, we briefly consider the design, operation and scheduling of a small batch process. By combining these problem aspects, we hope for synergies that lead, for instance, to shorter processing times, shorter planning horizons and higher quality batches. Two key aspects that aid in this case study are the use of simultaneous dynamic optimization and a simplified scheduling formulation. The former is due to the discretization of the differential equations (DAEs) and state and control profiles to form a large-scale optimization problem. For the latter we consider a continuous variable scheduling formulation adapted from Birewar and Grossmann (1986) that deals with the sequencing of tasks, products and equipment using idealized Unlimited Intermediate Storage (UIS) and Zero Wait (ZW) transfer policies. In both cases, a nonlinear program is formulated and the solution directly yields a minimum cycle time operating schedule for multiple products and stages. The detailed integrated formulation is given in Bhatia and Biegler (1996).

To demonstrate this approach we consider the process example given in Bhatia and Biegler (1996), which includes a batch reactor, batch distillation and several transfer units. We consider the sequencing of three products of varying purity with possible manipulations of the temperature profile in the reactor and the reflux profile in the column. Here optimal dynamic profile cases are R1 and C1 for the reactor and column, respectively, while best constant profiles are R0 and C0.

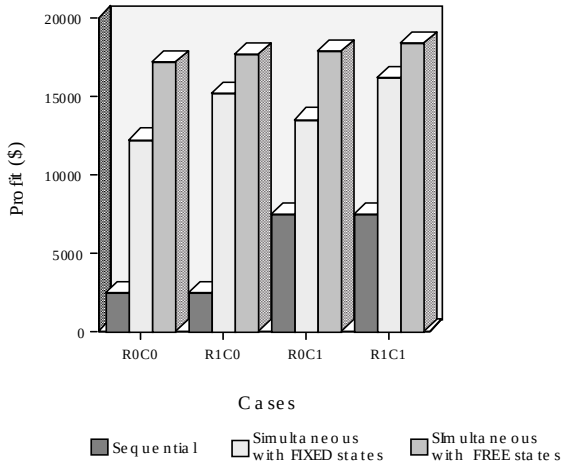


Figure 1: Integrated Batch Optimization

To compare formulations, we first consider the *sequential design* where the equipment design, operation and schedule are optimized one after another. The *intermediate case* deals with final time states fixed by a unit optimization followed by simultaneous optimization of design and scheduling. Finally, we consider a *fully simultaneous approach* with optimized final states. The results of this case study are summarized in Figure 1 for the ZW case. Note that significant improvements result from the integrated formulation. Here we see that there is strong improvement due to a variable reflux ratio (from C0 to C1) but the greatest improvements are simply due to the integration of the design, operation and scheduling tasks in the optimization problem. The reason for this is apparent from the production schedule. The sequential solution still has long cycle times and long slack times in the ZW schedule. The intermediate solution reduces both of these considerably and the fully simultaneous approach eliminates the slack times altogether and creates a schedule with the shortest cycle time. This allows for much greater equipment utilization and significantly improves the profit.

The message of this example is that key aspects of the dynamic operation, equipment design and scheduling need to be modeled accurately and optimized simultaneously. Without these features, even the most detailed unit optimizations (along with separately optimized schedules) would realize less than half of the benefits of the overall optimization, for this example.

Most of the above studies on integrated optimization were performed using open form models. However, as we shall see next, these models are not always appropriate. In the next section we contrast the benefits and drawbacks of open and closed form models. This sets the stage for alternative simultaneous strategies in Section 5.

4. Open Form vs. Closed Form Models

In process engineering, open form models (also known as equation oriented or equation based models) are characterized by a strategy whereby all of the stream and unit equations are assembled, solved and optimized as one large system. Additional calls to procedures are kept to a minimum, although procedural calls to physical property routines are still considered essential. There are several key advantages to this approach for process simulation and optimization. In particular, performance in solving large flowsheets can be accelerated considerably, and highly integrated flowsheets can be converged and optimized very efficiently. These models have been extremely successful in the real time optimization of petrochemical processes (Perkins, 1998; Marlin and Hrymak, 1997).

On the other hand, closed form models (also known as procedural, modular or black box models) are tightly integrated to a special purpose solver and the user can only access a restricted set of information from the model. As a result, optimization algorithms cannot access function and gradient information about individual model equations and variables; frequently the equations are not even represented explicitly within the model. Instead, in an optimization environment, such models are currently solved repeatedly and this leads to considerable computational cost. Closed form models are prevalent in process simulation, design and analysis, especially for detailed reaction and fluid flow models. In this section, we contrast these two approaches from a number of different perspectives.

Declarative vs. Procedural Modeling

The open form approach decouples modeling of the process from the solution algorithm. This allows a lot of freedom in formulating the model and also allows the user to concentrate on the proper problem definition for the task at hand. Moreover, it is conceptually easier to extend and modify declarative models because the solution procedure does not need to be modified at the same time. This declarative capability is the main strength of platforms like ASCEND, SPEEDUP, gPROMS and GAMS. Note that this is fundamentally different from a specific procedural model or even an EXCEL spreadsheet where the user is intimately focused on the procedure of solving as well as modeling.

This declarative form of modeling offers a number of advantages when extending and modifying closed form models. Provided that powerful procedures are available for this effort (including automatic differentiation of the model equations), this modeling strategy greatly shortens development time and provides for easier maintenance.

Implicit in the declarative approach is that a generic solver (usually, Newton-Raphson) will be able to solve the derived model. Moreover, this solver will take limited advantage of the physics of the model, except in the exploitation of sparsity in the linearized equations. This means that a number of specialized closed form algorithms that involve nested loops, bootstrapping solution strategies and interpolation to avoid expensive calculations are not straightforward to implement with open form modeling. These may be needed to avoid overly large systems, improve convergence or minimize expensive physical property calls (e.g., the inside-out algorithm in RADFRAC).

Newton-like Convergence Behavior

The open form approach allows for a simultaneous convergence strategy since the most efficient generic solver is Newton's method. Therefore, convergence should be Q-quadratic under ideal assumptions (Kantorovich, 1948). This means that properly formulated and initialized open form models can be solved one or two orders of magnitude faster than modular models. For problem formulations of comparable size, it is unlikely that an alternative approach can beat this performance. Moreover, this approach provides a general, coordinated convergence strategy, regardless of problem structure. This is in contrast to flowsheeting programs where convergence behavior strongly depends on the tear stream selection and unit sequencing.

On the other hand, Newton's method needs to be stabilized far from the solution, even with open form models. Here a generic stabilization strategy (line search or trust region) will not take advantage of specific problem features and the only recourse is for the user to reformulate the model. This becomes more serious when the problem becomes ill-conditioned or singular at isolated points. Under these circumstances, Newton's method has serious difficulties that should be tackled most effectively with problem-specific formulations. These can be remedied easily through additional safeguards in closed form approaches.

However, for open form models we need to resort to well-known line search and trust region strategies for solving nonlinear equations. In equation based environments these strategies are applied as a single damping factor for a trust region or line search. This can produce a 'conservative' stepsize that severely impacts overall convergence of the process system and offsets the expected quadratic convergence.

Toy Problem

To illustrate the problem of small stepsizes more closely, consider the following equations:

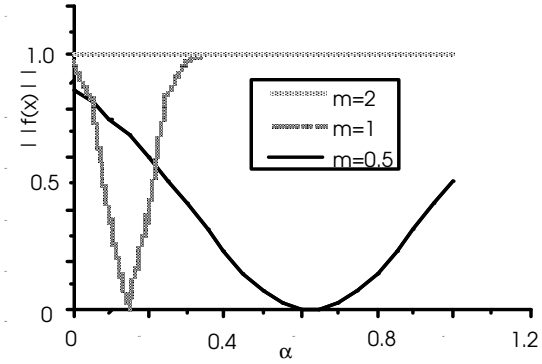
$$f_1 = 2\,x_1 + x_2 - 3 = 0$$

$$f_2 = \exp(-m(x_1-3)^2) - 1 = 0$$

$$x_1^0 = 1,\ x_2^0 = 1$$

First we consider the simultaneous solution of these equations with the above starting point using Newton iterations, for various values of m. As shown in Figure 2, a stepsize (α) equal to one is acceptable (using the Armijo criterion) for m = 0.5 at the first Newton step. However, for m = 5 only stepsizes below 1.3×10^{-7} will lead to a sufficient decrease in $\|f(x)\|$ along the first Newton direction. Figure 1 plots $\|f(x)\|$ for various values of m. Note that for m = 2 and above, $\|f(x)\|$ decreases only in very small regions around the ordinate. This will be true of *any open form model* that contains f_2. The table shows the number of MINPACK function evaluations to solve this problem. Here a very

reliable solver (using QR factorizations, trust regions and analytic Jacobians) needs to be used.



m	MINPACK	α_{max}	S-N
0.5	77	1.0	10
1.0	83	0.32	10
2.0	83	0.007	9
5.0	65	1.3e-7	12

Figure 2: Stepsizes for Toy Problem

In contrast, a procedural solution of this example is a trivial. Once x_1 is fixed, f_1 is solved immediately as a linear system. As shown above, f_2 can be solved using a simpler stabilized Newton method (S-N) where only a few iterations are needed. Of course, in both approaches, f_2 can easily be reformulated. But even with reformulation, the procedural approach offers the advantage of isolating the nonlinearities from the rest of the system.

The effect of the "conservative" stepsize has been observed on a number of process applications. For instance, simultaneous solution of countercurrent gas-gas heat exchanger models can be very unstable with very small damping factors required for convergence. Instead, a closed form reformulation of this system leads to a much more reliable method.

Open vs. Closed Form Structures

In addition to the advantages of declarative models, the open form of the model equations allows easy implementation and efficient calculation of exact derivative information. The ready availability of derivative information has led, along with Newton's method, to fast performance for open form approaches. Nevertheless, as discussed in the previous section, this can also be done in closed form using automatic differentiation strategies.

On the other hand, the open form necessarily requires the user to consider larger systems which are more difficult to set up, analyze, debug and get to solve. In essence, successful users *model declaratively but debug procedurally* by breaking down the problem and solving smaller problems. A closed form approach provides a natural, object oriented strategy for this analysis. As a result, the process solver must give the user access to smaller subproblems to accomplish this. This can also be done, in principle, with an open form approach (e.g., in the ASCEND system), but currently a lot more experience is required by the user to do this. Here most diagnostics deal directly with open form variables and equations, rather than diagnostic messages that relate to a physical model.

A related problem is the initialization of the open form model. This task requires substantial user intervention and often more time and effort than in obtaining the solution itself. For future development, the open form approach will need to be improved, not just with better user interfaces, but also with better tools that allow more flexible problem decomposition and reformulation.

Moreover, the fully open form approach is resistant to incorporating legacy codes in closed form. These may involve specialized process models or solution algorithms, which are essential for a particular application (e.g., CFD systems with an iterative solver). This is especially difficult for ODE and PDE-based models. A closed form approach can incorporate these specialized models into a larger process model much more readily than open form packages can. Often, these codes must be reformulated entirely as declarative models in the open form environment. In future development, we need to allow the integration of closed models that may provide derivative information, which will be used by the solver. An interesting development in this direction is the use of *foreign objects* in the gProms environment (Kakhu et al., 1998).

Finally, the issue of accurate discretization of models using differential equations in space must be dealt with. For instance, closed form models solve ordinary differential equations with automatic step sizes to control error. That is, the integration step size is not known *a priori* and usually varies from step to step. In the open form approach, it is not at all clear how these step sizes are to be chosen prior to discretization.

In academic studies, accurate discretization and error control has been demonstrated for small open form models (e.g., Tanartkit and Biegler, 1997). However, this may be difficult for commercial open form codes with many more stepsize variables. Currently, stepsize selection is done by trial and error by the modeler.

Conditional Models

With procedural approaches, it is straightforward to include IF-THEN relations triggered by inputs or calculated values. For instance, with simple flash calculations, decisions are based on dew and bubble point equations and the proper set of equations is then selected and solved. This approach is then embedded within much larger systems, which are successfully solved and optimized. The basic hope with closed form approaches, is that the solvers are able to 'jump over' these derivative and function discontinuities; this is often the case, even though it is easy to construct counterexamples.

There are more rigorous ways to deal with discontinuities and these are better suited to open form approaches. They are also more expensive than simple equation solving, as they involve combinatorial elements. Two examples are complementarity relations and integer variables. In both cases, the conditional relation must be reformulated by an experienced user to form a set of conditional equations or inequalities. The resulting model then must be solved with an advanced solver (nonlinear complementarity or mixed integer nonlinear program (MINLP), respectively).

Formulating Optimization Problems

Characteristics of open form modeling allow a natural extension of Newton-based solvers to nonlinear programming using algorithms like Successive Quadratic Programming (SQP), although it should be noted that *large-scale* versions of SQP are now required for simultaneous convergence. In contrast, much smaller optimization problem formulations can be considered for the same problem with the closed form approach.

Also, by extension, the same open form initialization effort required for model solution is needed for optimization. But, because more degrees of freedom are involved, the open form optimization may converge less reliably than the open form simulation problem. To promote convergence, safeguards that are easily built into the closed form model are not as straightforward in the open form approach.

Finally, the concept of formulating an optimization problem through the evaluation of procedures is often absent in the open form approach. In the modular approach, the user can visualize the optimization problem via a set of case studies applied to the (expected smooth) process model; these cases are then automated with an efficient, gradient-based (or even a direct search) optimization algorithm.

With proper safeguards, this automated case study approach leads to 'quick and dirty' optimization procedures which are often successful and lead to good, if not optimal, results. An open form approach to these problems requires complete rethinking of this concept and often results in a very different declarative model (e.g., MINLP formulation for feedtray optimization). In fact, the need to incorporate legacy codes and other closed form models has motivated several leading NLP researchers (Booker et al., 1998; Audet and Dennis, 1999; Kelley, 1999) and corporations (IBM, Boeing, Sandia Laboratories) to develop and analyze derivative free optimization methods.

5. Simultaneous Optimization with Existing, Closed-Form Models

From the previous section it is clear that both open and closed form models are required for different simulation and optimization applications. Still, a commonly held belief is that simultaneous gradient-based optimization is not possible with closed form models. As a result, there is considerable effort to extrapolate the open form successes from on-line optimization to many other areas of process engineering. As mentioned above, this effort will lead to a number of benefits in the creation, maintenance and efficient solution of many new process models. On the other hand, this approach does not incorporate legacy models well nor does it allow the use of (often essential) specialized solvers that exploit the model's structure.

Can we develop a simultaneous optimization approach using closed-form models? This section explores a hybrid strategy that allows the direct use of existing closed-form models within a simultaneous convergence and optimization strategy. Here we can also further exploit the

simultaneous strategy for closed form models with Newton-based solvers. Note that such models may still be difficult to represent in open form. They include complex column models, two point boundary value models that involve reaction, separation or heat transfer and the solution of nonlinear partial differential equations (PDEs). For these models we assume that *neither the equations nor the Jacobian matrix are available directly*. Instead, we expect to access the Newton step and the 'sensitivity' of this step to input variables. With this information we can derive the following optimization algorithm based on SQP concepts in the reduced space (rSQP).

Derivation of a Tailored rSQP Algorithm

Consider the Nonlinear Programming Problem (NLP):

$$\begin{aligned} &\text{Min } F(x) \\ &\text{s.t.} \quad h(x) = 0 \\ &a \le x \le b \end{aligned} \tag{1}$$

where $x \in R^n$ and $h(x)$: $R^n \dashrightarrow R^m$. To obtain a local optimum we need to find a point that satisfies Karush Kuhn Tucker (KKT) conditions. The first order KKT conditions can be given by:

$$\begin{aligned} &g(x^*) + \bar{h}(x^*)\,\lambda^* = 0 \\ &\bar{h}(x^*) = 0 \end{aligned} \tag{2}$$

where $g = \nabla F$ and $\bar{h}(x^*) = 0$ is the appropriate active constraint set at the solution x^*. To satisfy these equations (and choose the correct active set) we consider an extension of Newton's method by solving the QP subproblem at iteration k:

$$\begin{aligned} \text{Min} \quad & g(x_k)^T d + 1/2\, d^T Q\, d \\ \text{s.t. } & h(x_k) + A(x_k)^T d = 0 \\ & a \le x_k + d \le b \end{aligned} \tag{3}$$

where $A=\nabla h$, $Q = \nabla_{xx} L(x, \lambda)$ (the second derivative matrix of the Lagrange function with respect to x) or its approximation and d is the search direction for x. The solution of the QP is given by the linear system at iteration k:

$$\begin{bmatrix} Q & \bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \end{bmatrix} = - \begin{bmatrix} g \\ \bar{h} \end{bmatrix} \tag{4}$$

(the k subscript is suppressed for brevity). We now consider the exploitation of the structure of the QP. In many process optimization problems n ~ m >> (n-m). As a result, the n x n matrix Q can either be considered in a large, sparse form or approximated in the reduced space. Because second derivatives are not available for closed form models (or in most commercial packages that use open form models), we choose the latter option.

Here we develop a simplified decomposition in the reduced space using only equality constraints and defer the explicit treatment of variable bounds until later. To solve the QP in the reduced space, let:

$$x^T = [\, y^T \mid z^T \,], \quad A^T = [\, C \mid N \,]$$

and select an n x n nonsingular matrix:

$$H = [\, Y \mid Z \,], \text{ where } A^T Z = 0.$$

Here Z and Y form null and range space bases for the linearized equality constraints. We similarly partition the search direction into range (p_Y) and null space (p_Z) components: $d = Y p_Y + Z p_Z$.

The former component deals with the dependent variables of the model, while the latter component determines the search direction for decision variables of the optimization problem. With this representation we can write:

$$H = \begin{bmatrix} I & -C^{-1}N \\ 0 & I \end{bmatrix} \quad H^{-1} = \begin{bmatrix} I & C^{-1}N \\ 0 & I \end{bmatrix} \tag{5}$$

Defining the linear system from the QP as M x = f gives:

$$\begin{bmatrix} Q & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix} \tag{6}$$

Now defining X = diag [[Y | Z] , I], we can consider the equivalent system $X^T M X z = X^T f$ as:

$$\begin{bmatrix} Y^TQY & Y^TQZ & Y^TA \\ Z^TQY & Z^TQZ & 0 \\ A^TY & 0 & 0 \end{bmatrix} \begin{bmatrix} p_Y \\ p_Z \\ \lambda \end{bmatrix} = - \begin{bmatrix} Y^Tg \\ Z^Tg \\ h \end{bmatrix} \tag{7}$$

Because p_Y and p_Z are calculated from the last two rows, we can ignore the Y^TQY and Y^TQZ terms and approximate $Z^TQY\, p_Y$ by w_k to get:

$$\begin{bmatrix} 0 & 0 & Y^TA \\ 0 & Z^TQZ & 0 \\ A^TY & 0 & 0 \end{bmatrix} \begin{bmatrix} p_Y \\ p_Z \\ \lambda \end{bmatrix} = - \begin{bmatrix} Y^Tg \\ Z^Tg + w_k \\ h \end{bmatrix} \tag{8}$$

Note that neglecting these terms does not affect the search direction, only the multiplier estimates λ. As the search direction converges to zero, λ also converges to its correct value. Also we see that these assumptions lead to a simple block diagonal decomposition for the rSQP strategy.

To relate the range and null space representation back to the original variables, we define the above system as: M x = f, and define:

$$X = \text{diag}\,[\,[\,Z \mid Y\,]^{-1}, I\,].$$

This allows us to write $X^TM\,X\,z = X^Tf$ as:

$$\begin{bmatrix} 0 & 0 & C^T \\ 0 & B_k & N^T \\ C & N & 0 \end{bmatrix} \begin{bmatrix} d_y \\ d_z \\ \lambda \end{bmatrix} = - \begin{bmatrix} g_y \\ g_z + w_k \\ h \end{bmatrix} \tag{9}$$

where $B_k \sim Z^TQZ$. Now the C matrix is the combined Jacobian of the closed form multiple models and is assumed to be nonsingular. If not, we assume the closed form solver can be suitably modified to yield a nonsingular C. As a result we can modify the reduced system to yield:

$$\begin{bmatrix} 0 & 0 & I \\ 0 & B_k & N^TC^{-T} \\ I & C^{-1}N & 0 \end{bmatrix} \begin{bmatrix} d_y \\ d_z \\ C^T\lambda \end{bmatrix} = - \begin{bmatrix} g_y \\ g_z + w_k \\ C^{-1}h \end{bmatrix} \tag{10}$$

Note that if the model (h(x) = 0) is solved with a Newton-based procedure we may not be able to extract gradient information for the model equations. In many closed form models the Jacobian is not constructed explicitly and specialized decompositions are incorporated directly within the model (e.g., block tridiagonal decomposition in distillation models, condensation in collocation models). Nevertheless for many of these models, we can easily extract the Newton step, $p_y = - C^{-1} h$; the *model sensitivity matrix*, $D_z = - C^{-1} N$, which reflects the sensitivity of the Newton step to the model inputs can also be calculated. As a result, the KKT matrix for the QP is equivalent to:

$$\begin{bmatrix} 0 & 0 & I \\ 0 & B_k & -D_z^T \\ I & -D_z & 0 \end{bmatrix} \begin{bmatrix} d_y \\ d_z \\ \hat{\lambda} \end{bmatrix} = - \begin{bmatrix} g_y \\ g_z + w_k \\ -p_y \end{bmatrix} \tag{11}$$

and all of the explicit Jacobian information from the model disappears. Here the gradient of the objective function is usually specified directly by the user in terms of dependent and independent variables. The correction term w_k and the reduced Hessian B_k are approximated by reduced gradients. Moreover, D_z (the 'sensitivity' of p_y to decision variables, z) can be obtained by:

- calculating h/z within model and solving multiple right hand sides for - C^{-1} N
- perturbing h(x) with respect to z and solving multiple right hand sides for - C^{-1} N, or
- executing an additional Newton step with a perturbed value of z.

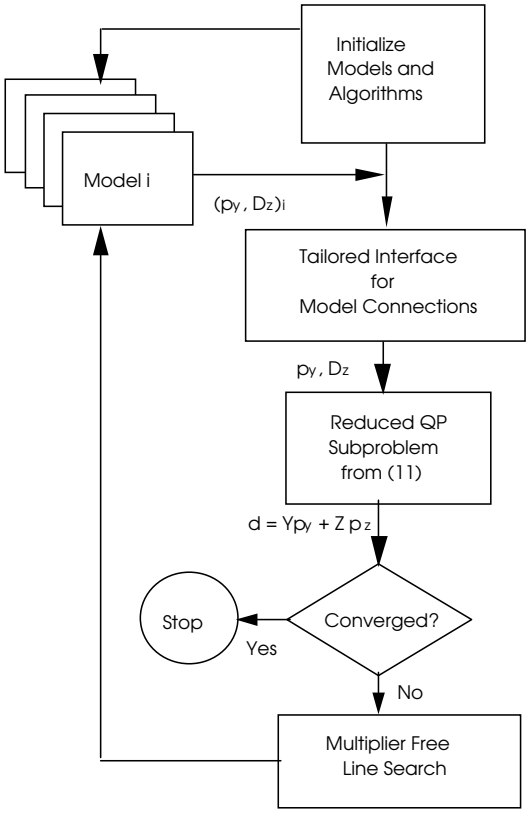


Figure 3:Tailored Process Optimization Algorithm

Given the Newton steps and model sensitivity matrices for each unit, it is easy to to create p_y and D_Z for the entire flowsheet, e.g., A" = [I |

D_Z]. The tailored rSQP algorithm can then be represented as in Figure 3. where the necessary model matrices are collected and decision variables are updated for the entire process. QPKWIK (Schmid and Biegler, 1994b) is used to solve the reduced space QP problem.

Finally, we see that the actual multipliers λ are not calculated; only $C^T\lambda$ is available and this precludes the use of conventional line search strategies for SQP. Instead we apply a 'multiplier-free' line search where the penalty parameter is estimated by a lower bound on the multipliers. A global and local convergence analysis related to this rSQP approach is given in Biegler et al. (1997). Examples on the application of this tailored approach to closed form models can be found in Schmid and Biegler (1994a) and Tanartkit and Biegler (1996). The details of extending the tailored approach to flowsheet optimization and related decomposition strategies are given in Alkaya et al. (1999).

Flowsheet Optimization Study

To illustrate this tailored approach we briefly summarize the flowsheet optimization study, described in Tanartkit and Biegler (1996). The process contains several distillation columns as well as an exothermic reactor model solved with a Newton based collocation routine (with fixed stepsizes). Here we compare a "simultaneous" closed form approach with the tailored optimization approach. In this closed form approach, the tear equations are converged efficiently as equality constraints in the optimization problem. Also, we found that an open form approach solved with an rSQP method and the same initialization performs about the same as the tailored approach.

Five cases were considered where shortcut (i.e., split fraction) models are substituted by detailed Newton based models for the reactor (using COLDAE) and distillation column (using UNIDIST) in the process flowsheet:

1. Product Column (UNIDIST)
2. Reactor Model (COLDAE)
3. Recycle Column (UNIDIST)
4. Both Columns (UNIDIST)
5. Reactor and Product Column (UNIDIST and COLDAE)

The performance of the tailored vs. the modular approach is shown in Figure 3. In each of these cases the tailored approach leads to far better performance than in the conventional closed form (modular) approach. In the last case where both models need to be considered, the tailored approach leads to a four-fold increase in performance, but without requiring open form models to be used.

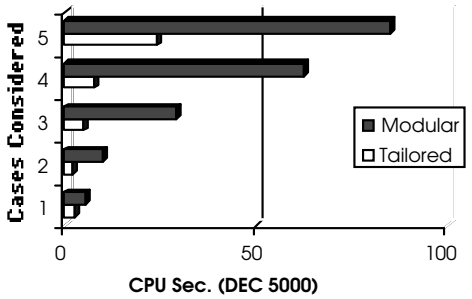


Figure 4. Flowsheet Optimization with the Tailored Approach

This approach has also been extended to dynamic systems as described in Alkaya et al. (1998). Here an optimal control problem for a binary batch column model is coupled to a batch reactor. Both units are self-contained collocation models and coupling these models over time is performed using the above decomposition procedure. The optimization of this system with the tailored approach generally leads to the same performance as with open form models. Here finding the optimal reflux ratio to maximize product requires about 29 rSQP iterations.

6. Further Challenges for Nonlinear Programming Algorithms

The tailored approach represents only a preliminary effort to exploit models with closed form structure and multiple solvers. In this section we review a number of essential research issues for large-scale nonlinear programming.

Handling bound constraints

The development of the tailored approach does not explicitly deal with bound constraints. These have been handled in many process optimization applications (for both open form and tailored models) by the rSQP subproblem shown in Figure 3. For our studies we have relied on a reduced space QP algorithm (QPKWIK) to handle inequalities but this has clear limitations for larger problems. To deal with large-scale decomposition strategies, the KKT matrix should be exploitable and should not interfere with updating of bound constraints. This

can be done by exploring both active set and interior point (IP) methods.

On problems where (n-m) is large (e.g., optimal control problems and multiperiod problems) interior point methods have a tremendous performance advantage in active set selection (Albuquerque et al., 1999). Figure 5 compares the number of active set iterations with the interior point method. Note that the number of IP iterations (six) remains independent of (n-m). Solving QP subproblems with an IP strategy has yielded significant benefits for process applications in process identification, multiperiod optimization and model predictive control.

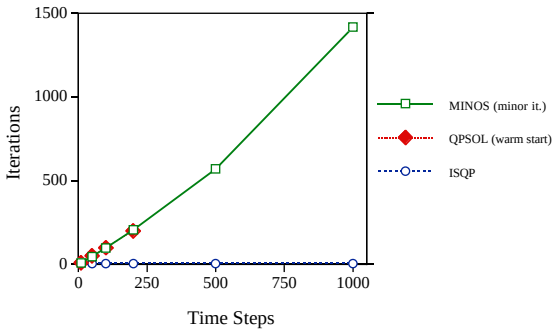


Figure 5. Comparison on Interior Point (ISQP) vs. Active Set (QPSOL, MINOS) iterations when (n-m) is large

To increase the usefulness of this approach we are currently exploring IP methods to solve the NLP problem directly. This further eliminates the overhead of solving the QP with an IP approach and can lead to an order of magnitude performance improvement. This approach has seen much recent work (Vanderbei and Shanno, 1999; Byrd, Gilbert and Nocedal, 1998; Conn, Gould and Toint, 1999) but many open questions still remain. Nevertheless, preliminary results show this approach is very successful for optimal control problems. For instance, with an IP NLP approach, the dynamic optimization of an air separation column with 462 DAEs and over 55,000 variables requires less than 5 CPU minutes on a DEC Alpha workstation (Cervantes et al., 1999).

On the other hand, for problems where (n-m) is small, the number of active bounds and the search complexity are also small. Therefore the overhead associated with IP methods does not compete with efficient active set methods (Ternet and Biegler, 1999). In particular, the application of Schur complement methods (Betts and Frank, 1994) has the advantage that the KKT matrix can be decomposed separately and updates to the active set are very cheap. This approach can be applied directly to (10) or (11) and appears to be about three times faster than QPKWIK (Bartlett, 1999).

Iterative Linear Solvers

Similar concerns in exploiting the structure of the KKT matrix arise when dealing with very large NLPs arising from PDE applications. Large finite element models for fluid flow or structures are also solved with Newton-based procedures but are too large to attack with direct, sparse linear solvers. Instead iterative linear solvers (Preconditioned Krylov (PK) methods) are essential and the extension to simultaneous SQP methods poses a severe challenge. Recently, Biros and Ghattas (1999) used the rSQP decomposition and an approximated reduced Hessian to precondition the full space KKT matrix; the PK solver MINRES was used in a parallel environment to solve the KKT system. For a 3-D finite element model of the Navier-Stokes equations (with 90,000 state variables, 1200 decision variables) they obtained optimal solutions with less than 200 wallclock secs on Cray T3E. This led to an average reduction of 34 times over rSQP method mentioned above and showed excellent scalability of the method on parallel processors. Moreover, on problems with few decision variables, Ghattas and coworkers have applied rSQP concepts to optimize finite element models with over 3 million state variables.

In addition to the above research directions, there are a number of specific issues related to handling increasingly large process optimization problems. Here we also need to consider:

Decomposition of large nonlinear systems for problems that exhaust the limitations of current tools. Included here are the handling of (multiple) RTO models for refinery wide optimization and multiple plant design models, which may run on different platforms, exploit different structure and could be developed under different modeling environments.

Coupling of multiple design models where mass and energy balance models need to be coupled with detailed equipment design and costing. Also included are important process aspects such as operability, controllability and safety, which can be hard to formulate for the optimization problem (often due to discontinuities in their calculation).

Multiperiod models reflect life cycle considerations for the process and allow the design to consider different operating scenarios, consideration of

uncertainty and evolution of the process over time. These models lead to very large bordered block diagonal (or almost block diagonal) KKT matrices, where each diagonal block could represent the KKT matrix of any of the process optimization problems discussed thus far.

Most studies that respond to the challenges in integration assume that all process models are available in an open form environment. In this environment, discontinuities and nonsmoothness can be overcome, in principle, through additional constraints or MINLP formulations.

However, if process models need to remain in closed form, can the barriers to optimization be overcome? Here we can handle nonsmooth problems through reformulation or direct handling with bundle methods. More severe problems have prompted recent activity in derivative free optimization (DFO) methods in the math programming community, even to solve MINLP problems (Audet and Dennis, 1999). Also, convergence properties that link gradient based methods and DFO methods for unconstrained optimization, have recently been analyzed by Kelley (1999). In performance, these methods are clearly inferior to gradient based methods used with open form models. However, their robust application to closed form models could avoid the daunting expense of model conversion. Given the resurgence of these methods, an open question is whether they can be extended rigorously to constrained optimization problems with simultaneous convergence.

Finally, recent progress in the development of powerful methods for global optimization and their application to process problems (see Floudas and Pardalos, 1999) raises similar questions about their applicability to closed form models. It also focuses an awareness of the importance of local vs. global solutions in practice and better appreciation of the capabilities of optimization tools.

7. Conclusions

The widespread diversity of process models needs to be recognized and their individual structure needs to be exploited both for integration and optimization of these models. In the case of open form (equation based) models, extremely efficient optimization algorithms have been developed. Their influence is felt in the rapid solution of large-scale process optimization models, particularly for real-time optimization. Moreover, this capability has led to very creative NLP and MINLP problem formulations that allow engineers to model and exploit a number of difficult process attributes (e.g., integration of operability, controllability, safety constraints) in an efficient, quantitative manner.

However, more complex models need to be considered for process optimization and integration. And not all of these models can be represented in an open form, to be solved with a single generic solver. Difficult nonlinear features need to be isolated, particular equation structures need to be exploited more efficiently and specialized unit solvers are essential for many applications. To deal with these kinds of models, we outline a tailored optimization approach and illustrate its simultaneous capabilities on both flowsheeting and DAE models. This approach represents only a preliminary step toward exploiting these models. Moreover, a number of emerging areas in the optimization community are lending new algorithms and fundamental concepts to deal with these problems.

Finally, the ability to deal with specialized models is important not only for performance but also for the challenges posed for the integration of multiple process models, process attributes and for corporate-wide activities including design, control, operation, scheduling and planning. Application of optimization strategies is a key component of this integration and the most benefits are obtained *only if* accurate process models can be considered in the overall problem formulation.

Acknowledgements

Research funding from Mobil Research Corporation, Air Products and Chemicals, Inc. and the National Science Foundation is gratefully acknowledged. The authors would also like to thank Omar Ghattas, Wolfgang Marquardt, Peter Piela, Purt Tanartkit, Dimitrios Varvarezos and Sriram Vasantharajan for helpful discussions and comments.

References

Abel, O., A. Helbig, W. Marquardt, "Productivity Optimization of an Industrial Semi-batch Polymerization Reactor under Safety Constraints" Journal of Process Control, to appear, 1999.

Albuquerque, J., V. Gopal, G. Staus, L. T. Biegler, and B. E. Ydstie, "Interior Point SQP Strategies For Large-scale, Structured Process Optimization Problems," *Computers and Chemical Engineering* , 23, p. 543 (1999)

Alkaya, D., and S. Vasantharajan, and L. T. Biegler, "Generalization and Application of Tailored Approach for Flowsheet Optimization," Annual

AIChE Meeting, Miami Beach, FL, November, 1998

Alkaya, D., and L. T. Biegler, "Exploiting Chemical Process Model Structure for Nonlinear Programming", SIAM Annual Meeting, Atlanta, 1999

Audet, C. and J.E. Dennis, "On the Convergence of Mixed Integer Pattern Search Algorithms, Tech. Report, CAAM, Rice University (1999)

Backx, T., O. Bosgra, W. Marquardt, "Towards Intentional Dynamics in Supply Chain Conscious Process Operations" FOCAPO '98, Snowbird, Utah, July 5-10, 1998.

Balakrishna, S. and L. T. Biegler, "Chemical reactor network targeting and integration - an optimization approach," Advances in Chemical Engineering, 23, p. 248 (1996)

Bartlett, R. A., PhD Proposal, Carnegie Mellon University, Pittsburgh, PA (1999)

Betts, J. T., "What's the Answer? An Industrial Perspective on Optimization," presented at SIAM Conference on Optimization, Atlanta, GA, May, 1999

Betts, J. T., and P. Frank, "A sparse nonlinear optimization algorithm," *J. Opt. Theo. Appl*, 82, 3, (1994)

Bhatia, T., and L. T. Biegler, "Dynamic Optimization in the Design and Scheduling of Multiproduct Batch Plants," *I & EC Research*, **35**, 7, p. 2234 (1996)

Bhatia, T. and L. T. Biegler, "Multiperiod Design and Planning with Interior Point Methods," *Computers and Chemical Engineering*, accepted for publication (1999)

Biegler, L.T. and R.R. Hughes, "Approximation Programming of Chemical Processes with Q/LAP," Chemical Engineering Progress , p. 76, April (1981).

Biegler, L.T. and R.R. Hughes, "Infeasible Path Optimization for Sequential Modular Simulators," AIChE J. 28, 6, p. 994 (1982).

Biegler, L. T., Claudia Schmid, and David Ternet, "A Multiplier-Free, Reduced Hessian Method For Process Optimization," *v. 93*, p. 101, IMA Volumes in Mathematics and Applications, Springer Verlag (1997)

Birewar, D. and I. E. Grossmann, "Efficient algorithms for the scheduling of multiproduct batch plants," I & EC Research, 28, p. 1333 (1989)

Biros, G. and O. Ghattas, "Parallel domain decomposition methods for optimal control of viscous incompressible flows," Proceedings of Parallel CFD '99, Williamsburg, VA (1999)

Booker, A. J., Paul D. Frank, J.E. Dennis, Douglas W. Moore, and David B. Serafini, "Managing Surrogate Objectives to Optimize a Helicopter Rotor Design - Further Experiments," Tech. Report, CAAM, Rice University (1998)

Byrd, R. H., J-C. Gilbert and J. Nocedal, "A Trust Region Method based on Interior Point Techniques for Nonlinear Programming," Technical Report, ECE Dept., Northwestern University (1998)

Cervantes, A., A. Waechter and L. T. Biegler, "A Decomposed Barrier Strategy for Optimal Control" presented at SIAM Conference on Optimization, Atlanta, GA, May, 1999

Chen, H-S., M. A. Stadtherr, "A simultaneous modular approach to process flowsheeting and optimization - I, II, III," *AIChE J.*, 31, 11, p. 1843 (1985)

Conn, A. R., N. Gould, D. Orban, "Global and Local Convergence of a Primal-Dual Barrier Algorithm for Nonlinear Optimization," presented at SIAM Conference on Optimization, Atlanta, GA, May, 1999

Duran M. A., and I. E. Grossmann, "Simultaneous optimization and heat integration of chemical processes," AIChE J., 32, p. 123 (1986)

Feehery, W. F., J. Tolsma, P. Barton, "Efficient sensitivity analysis of large-scale DAEs," *Appl. Num. Math.*, 25, p. 41 (1997)

Floudas, C. A. and C. Pardalos, "Recent Developments in Global Optimization Methods and their Relevance to Process Design," presented at FOCAPD '99 (1999)

Friedman, P. and K. L. Pinder, "Optimization of a Simulation Model of Chemical Plants," *I & EC Proc. Des. Dev.*, 11, 4, p. 512 (1972)

Gaines, L. D. and J. L. Gaddy, "Process Optimization by Flowsheet Simulation," *I & EC Proc. Des. Dev.*, 15, 1, p. 206 (1976)

Griewank, A., "On Automatic Differentiation," *Mathematical Programming: Recent Developments and Applications*, eds. M. Iri, K. Tanabe, Kluwer Academic Publishers, pp. 83-108, 1989.

Grossmann, I. E., & J. Hooker, "Logic Based Approaches for Mixed Integer Programming Models," presented at FOCAPD '99 (1999)

Helbig, A., O. Abel, W. Marquardt, "Structural Concepts for Optimization Based Control of Transient Processes," Proceedings of Nonlinear Model Predictive Control Workshop, Ascona, Switzerland, 1998.

Ierapetritou, M.G.; Pistikopoulos, E.N.; Floudas, C.A., "Operational planning under uncertainty, " Comput. Chem. Eng., 20, 12 (1996) pp. 1499-516

Jensen, K. F., "Integrated Microchemical Systems: Opportunities for Process Design," presented at FOCAPD '99 (1999)

Kaijaluoto, S., "Process optimization by flowsheet simulation," Technical Research Center of Finland, Publ. #20 (1984)

Kakhu, A., B. Keeping, Y. Lu and C. Pantelides, "An Open Software Architecture for Process Modeling and Model-Based Applications," FOCAPO Proceedings, *AIChE Symposium Series # 320*, p. 518 (1998)

Kantorovich, L. V., "Functional Analysis and Applied Mathematics," *Uspekhi Mat. Nauk*, 3, p. 89 (1948)

Kelley, C. T., *Iterative Methods for Optimization*, SIAM (1999)

Li, S., L. Petzold and W. Zhu, "Sensitivity analysis of differential-algebraic equations: a comparison of methods on a special problem," Tech. Report, Computer Science Department, University of California, Santa Barbara (1999)

Locke, M.H.; Westerberg, A.W., "The ASCEND-II system-a flowsheeting application of a successive quadratic programming methodology," *Computers & Chemical Engineering*, 7, 5, p. 615 (1983)

Logsdon, J. S. and L. T. Biegler, "Accurate determination of optimal reflux policies for the maximum distillate problem in batch distillation," I & EC Research, 32, 4, p. 692 (1993)

Luyben, M.L.; Floudas, C.A., "Interaction between design and control in heat-integrated distillation synthesis," Integration of Process Design and Control (IPDC '94) IFAC Workshop, Oxford, Elsevier; vii+251; 1994; pp. 79-84

Marlin and Hrymak, "Realtime operations optimization of continuous processes," Chemical Process Control V, J C Kantor, C E Garcia and B Carnahan (eds.) p. 217 (1997)

Marquardt, W. "Perspectives on Life Cycle Process Modeling," presented at FOCAPD '99 (1999)

Marquardt, W., and M. Nagl, "Tool Integration via Interface Standardization?" Computer Application in Process and Plant Engineering - Papers of the 36th DECHEMA-Monographie 135, 1998, 95-126.

Morari, M. and J. Perkins, "Design for Operations," FOCAPD '94, CACHE Corporation, Biegler and Doherty (eds.), p. 105 (1994)

Mujtaba, I. and S. Macchietto, "Optimal operation of multicomponent batch distillation- multiperiod formulation and solution," *Comp. Chem. Engr.*, 17, 12, p. 1191 (1993)

Parker, A. L. and R. R. Hughes, "Approxiamtion Programming of Chemical Processes - I," *Comp. Chem. Engr.*, 5, 3, p. 123 (1981)

Perkins, J., "Plant Wide Optimization - Opportunities and Challenges," FOCAPO '98, Snowbird, Utah, July 5-10, 1998.

Pinto, J.M., Grossmann, I. E., "Optimal cyclic scheduling of multistage continuous multiproduct plants," Comput. Chem. Eng. 18, 9, pp. 797-816 (1994)

Pistikopoulos, E., "Uncertainty in process design and operations," Comp. Chem. Engr., 19, p. S553 (1997)

Ramage, M. "Computation and Competitiveness: Managing Technology in the Information Age," FOCAPO '98, Snowbird, Utah, July 5-10, 1998.

Sahinidis, N. V. and I. E. Grossmann, "Multiperiod investment model for process networks with dedicated and flexible plants," I & EC Research, 30, 6, p. 1165 (1991)

Schmid, C. and L. T. Biegler, "Quadratic Programming Methods for Tailored Reduced Hessian SQP," *Computers and Chemical Engineering*, **18**, 9, p. 817 (1994a)

Schmid, C. and L. T. Biegler, "A Simultaneous Approach for Process Optimization with Existing Modeling Procedures," *Chemical Engineering Research and Design, Part A*, **72**, p. 382 (1994b)

Shanno, D., and R. Vanderbei, "An Interior-Point Algorithm for Nonconvex Nonlinear Programming," presented at SIAM Conference on Optimization, Atlanta, GA, May, 1999

Sinclair, J. L., "CFD for Multiphase Flow: Research Codes and Commercial Software" presented at FOCAPD '99 (1999)

Stephenson, G. R., and C. F. Shewchuk, "Reconciliation of Process Data with Process Simulation," *AIChE J.*, 32, 2, p. 247 (1986)

van Schijndel, J. and S. Pistikopoulos, "Towards the Integration of Process Design, Process Control and Process Operability - Current Status and Future Trends," presented at FOCAPD '99 (1999)

Tanartkit, P., and L. T. Biegler, "Reformulating Ill-Conditioned DAE Optimization Problems," *I & EC Research*, **35**, 6, p. 1853 (1996)

Tanartkit, P. and L. T. Biegler, "A Nested, Simultaneous Approach for Dynamic Optimization Problems-II: The Outer Problem," *Computers and Chemical Engineering*, **21**, 12, p. 1365 (1997)

Ternet, D. and L. T. Biegler, "Interior Point Methods for Reduced Hessian Successive Quadratic Programming," *Computers and Chemical Engineering*, accepted for publication (1999)

Voudouris, V. T., and I. E. Grossmann, "Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory consideration," I & EC Research, 32, 2, p. 1962 (1993)

Walsh, S. and J. Perkins, "Application of integrated process and control system design in waste water neutralization," Comp. Chem. Engr., 8, p. S183 (1994)

Wolbert, D., X. Joulia, B. Koehret and L. T. Biegler, "Flowsheet Optimization and Optimal Sensitivity

Analysis Using Exact Derivatives," *Computers and Chemical Engineering*, **18**, 11/12, p. 1083 (1994)

Wright, S. "Algorithms and Software for Linear and Nonlinear Programming," presented at FOCAPD '99 (1999)