Tailoring Optimization Algorithms To Process Applications

L T. Biegler

EDRC 06-132-92

TAILORING OPTIMIZATION ALGORITHMS TO PROCESS APPLICATIONS

Lorenz T. Biegler

Chemical Engineering Department, Carnegie Mellon University
Pittsburgh, PA 15213 USA

Over the past decade the application of efficient nonlinear programming tools has become a powerful and established strategy for process analysis and design. While simple coupling of standard optimization codes to process applications has demonstrated the effectiveness of this approach, further development is needed for the efficient *large-scale* use of these tools. This is particularly true for dynamic optimization problems and on-line applications, as well as design problems with complex models.

This paper centers on exploiting the structure of process optimization problems. Here we consider the tailoring of Newton-type optimization algorithms for various process applications. In particular, the Successive Quadratic Programming (SQP) algorithm has been successful over the past decade; it has also been extended in a number of ways to optimization problems involving several thousand variables. After describing a large-scale, general purpose NLP solver, we briefly describe specialized SQP algorithms for die following problem classes:

- optimization of complex, structured process models with few degrees of freedom

- optimization of dynamic systems, where structured sets of nonlinear equations are created from discretization of differential-algebraic systems (DAEs)

- treatment of parameter estimation problems with many degrees of freedom

Each of these aspects is illustrated with a small process application and efficient techniques for handling these problems are outlined. Also, these methods are briefly compared to general purpose algorithms in order to demonstrate the effectiveness of this approach.

INTRODUCTION

Since the application of the Successive Quadratic Programming (SQP) algorithm in the late seventies there has been renewed interest in process optimization as a design and analysis tool. Based on the work of Han (1977) and Powell (1977), several variants of this algorithm have been developed for flowsheet optimization, both with equation-based (Berna et aL, 1980; Locke et al., 1983) and modular process simulators (Biegler and Hughes, 1982; Chen and Stadtherr, 198S). Today, flowsheet optimization is an established design tool and at least a dozen commercial process simulation programs incorporate an SQP optimization strategy. However, given the acceptance of process optimization, the next challenge is to maintain the reliability and efficiency of the optimization approach as more difficult design problems are considered. Part of this challenge lies in the education of the design engineer, who needs to develop an appreciation of the characteristics and the limitations of these tools. This will lead to more attention to the proper, intelligent formulation of optimization problems and an interactive approach to the application of optimization tools (e.g., Amargerct aL, 1991). On the other hand, general nonlinear programming (NLP) algorithms such as SQP need to be tailored to address larger; structured process design problems efficiently, as these are being considered more frequently by industry. This paper considers this latter question and explores three general problem classes for which SQP has been tailored. Specific instances are given for each of these classes and process examples arc presented to demonstrate the effectiveness of this approach.

First, we explore the extension of SQP to complex, but structured, process models. Here, a general decomposition method is presented that efficiently deals with large, nonlinear models with few degrees of freedom. These are characteristic of most complex design problems in process engineering. If, in addition, the process model is solved with a tailored, Newton-Raphson algorithm that exploits the problem structure (e.g., block banded systems for separation units) we show that the application of SQP decomposition leads to an efficient optimization algorithm that can be implemented as a straightforward extension to the model solver. A comprehensive outline of this approach is given in the next section along with an example that exploits the Naphthali-Sandholm distillation algorithm.

Following this, we consider process models composed of a system of differential-algebraic equations (DAEs). Here we consider **a** flexible formulation where the DAE system is discretized and incorporated directly into the formulation of **the** optimization problem. This "simultaneous*[1] **approach leads** to combined **solution of both the model and the optimization problem and allows for the direct treatment of profile constraints. However, the larger size of the resulting system dictates the application of structured decomposition strategies if complex and realistic process models are to be considered. Here, a tailored strategy is outlined and applied to the dynamic optimization of a nonisothermal batch reactor. The fourth section deals with problem formulations that arise in data reconciliation and parameter estimation. As in the previous two sections, the** structure of the process **model can be incorporated into the decomposition procedure. Also,** the **least squares structure** of **the optimization problem leads to a tailored quasi-Newton strategy. Application of these concepts to SQP leads to a parameter estimation algorithm that is over twice as fast as the general purpose method. Moreover, certain classes of data reconciliation and parameter estimation problems have large degrees of freedom due to underdetermincd models. Here the effort of general purpose nonlinear programming algorithms increases polynomially with problem size (i.e. number of data sets). On the other hand, we show that by exploiting the special structure of these problems a tailored SQP algorithm is developed with performance that increases only linearly with the number** of **data sets.**

The final section of the paper summarizes these concepts and briefly addresses some open questions. Clearly, an important issue lies in increasing the number and breadth of tailored process applications. To this end, modular and flexible optimization implementations are required along with appropriate interfaces to large-scale modelling tools. In addition, a flexible approach to understanding the structure of process models needs to be adopted so that proper formulation of the process optimization problem includes the rapid development of special-purpose nonlinear programming algorithms.

HOW SQP CAN BE TAILORED

In this section we first develop a large-scale SQP algorithm that can be tailored to solution algorithms for large-scale process models. Here we consider, for simplicity, the following general nonlinear programming formulation:

$$\text{Min } \Phi(z)$$

$$\text{s.t. } h(z) = 0, \qquad h: \Re^n \rightarrow \Re^m \qquad (1)$$

$$z^L \le z \le z^U$$

Problem (1) can be solved by any large scale nonlinear programming solver (e.g., MINOS, Successive Linear Programming, GRG, etc.). For this work, we consider Successive Quadratic Programming (SQP) for the optimization step. SQP generally requires fewer function evaluations

than other solvers and has been shown to be robust for nonlinear programs that arise from complex process models. Motivated by a Newton-based approach applied to the optimality conditions of (1), SQP solves a quadratic programming (QP) subproblem in order to update the variables z. The QP subproblem of problem (NLP) at the kth iteration is given by:

$$\min_{d} \quad \nabla\Phi^T(z_k)\, d + \frac{1}{2} d^T B^k d$$

$$\text{s. t.} \qquad (2)$$

$$h(z_k) + Vh^T(z_k)\, d = 0$$

$$z^L \leq z_k + d \leq z^u$$

where

d : search direction vector

B^: analytical or approximated Hessian **of the** Lagrange function

and necessary conditions of (2) (with bounds inactive) **arc given by**

$$\begin{bmatrix} B & Vh \\ Vh^T & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = - \begin{bmatrix} \nabla\Phi \\ h \end{bmatrix} \qquad (3)$$

Here the index k is suppressed and Vh is the [n by m] matrix of equality constraint gradients. As problem (1) becomes large, the "vanilla" SQP algorithm becomes inefficient since B is a dense n x n matrix and no advantage is taken of model sparsity. One approach to overcome this limitation is to develop sparse implementations (Nickel and Toilc, 1989; Lucia and Xu, 1990), but care needs to be taken to ensure global convergence. It should be noted, though, that these approaches can be very successful for large-scale applications. Another approach deals with decomposition of (3) and updating projections of B onto a space characterized by the independent variables of the problem. For our applications we take advantage of these decomposition approaches as Che number of independent variables for process optimization is relatively small (10 to 100) and the actual projected Hessian is expected to be small, dense and positive definite.

To develop the decomposition, let Z(zfc) be an [n by (n-m)] matrix with columns spanning the null space of $Vh^T(zfc)_f$ and let Y(zfc) be an [n by m] matrix so that [YZ] **is nonsingular. Let Q be a nonsingular matrix of order (n+m), given by:**

$$Q = \begin{bmatrix} [YZJ & 0 \\ 0 & I \end{bmatrix} \qquad (4)$$

where I is an m-dimcnsional identity matrix. Then the search direction can be written as:

$$d = [V\ Z] \begin{bmatrix} d_y \\ d_z \end{bmatrix} = Yd_y + Zd_z. \qquad (5)$$

and multiplying equation (3) by Q and substituting for d yields:

$$\begin{bmatrix} Y^TBY & VBZ & Wh \\ Z^TBY & ZTBZ & 0 \\ Vh^TY & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} Y^T\nabla\Phi \\ pv<t> \\ h \end{bmatrix} \qquad (6)$$

Thus the QP (2) has the following solution:

$$d_y = -(Vh^TYY^l\ h$$
$$d_2 = -(Z^TBZ)\text{-}^! (Z^T\ VO + Z^TBY\ d_y)$$
$$v = -(Y^T\ Vh)\text{-}^1 (Y^T\ V{<}D + Y^TBd)$$

and d is reconstructed from equation (5). Since the search direction goes to zero at convergence, the multiplier estimates can be approximated by

$$v = -(Y^T\ \nabla h)^{-1}\ Y^T\ \nabla\Phi$$

and the matrix Y^BY need not be estimated. Based on this decomposition, the reduced QP subproblem with bounds is given by:

$$^Mj£ \qquad (V{<}D(z_k) + BYd_y)^T\ Z\ d_z + \frac{1}{2} dj\ (Z^TBZ)\ d_z \qquad (7)$$
$$\text{s.t.}$$
$$z^L\ £\ Zk + Zd_z + Yd_y\ £\ z^u$$

Note that the projected Hessian (Z^BZ) is usually positive definite and a dense, positive definite quasi-Ncwton approximation (e.g. the BFGS update) is a good match for this structure. Often, the projected Hessian in the range space, Z^BY, is neglected since it is assumed that dy is small. Neglecting Z^TBY leads to a 2-step q-superlinear rate of convergence (Nocedal and Overton, 1985), which is often acceptable for process optimization. Alternatively this term can be estimated by finite difference of the reduced gradients along Ydy, or approximated by a Broyden update of Z^TB. Both approaches can improve the rate of convergence.

In addition, there are several choices for the basis matrices, Y and Z. In the math programming literature, orthonormal bases, created from QR factorizations of Vh, are applied (see Nocedal and Overton, 1985). However, this dense decomposition is expensive and not practical for solving large problems. Instead, the following *orthogonal* representations are often used (see Vasantharajan and Bieglcr, 1988):

$$Z^T = [I \quad -A^T] \qquad \text{and} \qquad Y^T = [A \quad I] \qquad \text{where} \quad A = (V_xh^TY^l\ V_uh^T.$$

Here u and x represent the independent and dependent variables (of z), respectively. This basis takes advantage of sparsity in Vh^T but requires a nonsparse "least squares" step, dy, from (6). Note that $Y^TZ = 0$ so the step dy is minimized. Gabay (1982) and Locke et al. (1983) suggested setting $Y(z)^T = [\ 0 \quad I\ ]$, i.e., along a *coordinate* direction or basis for a set of dependent variables. From (6) we see that this basis has the particular advantage in that the step dy comes directly from a Newton step for solving h(x) = 0. Consequently, the reduced Hessian SQP method with coordinate bases is straightforward to tailor to structured Newton-Raphson solution procedures for complex models. However, Vasantharajan and Biegler (1988) and Schmid and Biegler (1991) have observed that because dy can become large and curvature information along this direction (Z^TB Y) is ignored, *coordinate bases* can lead to less reliable SQP algorithms. To remedy this, the latter study provides estimates of Z^TB Y dy and shows that slow convergence problems can indeed be overcome, especially in cases where coordinate bases normally fail. In addition, Biegler et al. (1991) develop an adaptive procedure for estimating this missing term that leads to l-?tep superlinear convergence.

As an illustration, Schmid and Biegler apply this approach to the Naphthali-Sandholm distillation model, UNIDIST, that is part of die SEPSIM process simulator (Andersen, ct al., 1991). Here the model has a block tridiagonal structure in the dependent variables (flowratcs) and an efficient Thomas algorithm is applied to obtain the Newton step for the distillation model. This program was modified and extended in a straightforward manner to include distillation optimization. The data structures for the modelling equations and the tridiagonal system remain virtually the same and none of the physical property routines were modified. In addition, only a few changes were made to the interface in order to allow a flexible description of the optimization problem. In this way, the existing model is coupled to the reduced SQP method and converted into a tailored optimization method for distillation. As a simple example, Schmid and Biegler consider a 12 tray column for separating benzene and toluene. Using the reflux ratio and column pressure as decision variables, a weighted objective involving product rate and reboiler duty is minimized. Here solution of this optimization problem with a nested approach (and repeated solution of the distillation model) takes seven SQP iterations and 90 Newton iterations for the column model. On the other hand, the simultaneous optimization approach, starting from the column's default initialization requires only 18 **SQP** iterations with coordinate bases (with finite difference estimates of Z^TBY dy) and 21 iterations with orthogonal bases. Here, the SQP optimization is only about twice as expensive as the Newton-Raphson column simulation.

STRUCTURES OF DAE SYSTEMS

To **extend** this approach to models described by systems of differential-algebraic equations, consider the following dynamic optimization problem for t £ [a,b]

$$\underset{u(t),x(t),\,p}{\text{Min}} \quad *F(x(b),p) + \int_a^b G(x(t),u(t),p)\;dt \qquad (8)$$

s.L

$$\dot{x}(t) = F(x(t),u(t),p)$$

$$g(u(t),x(t)) \;£\; 0$$

$$g_f(x(b)) \;£\; 0$$

$$x(a) = xo$$

$$x(t)^L \;£\; x(t) \;£\; x(t)U$$

$$u(t)^L \;£\; u(t) \;£\; u(t)^U$$

where:

$*F(x(b))$ • component of objective function evaluated at final conditions

$\int_a^b G(x(t),u(t),p)\;dt$ **= component of objective function over time**

g = inequality design constraint vector
x(t) = state profile vector
u(t) = control profiles
p = design parameters, not time dependent
gf = inequality constraints at final conditions
xn = initial condition **for state vector**
xL(t), $x^u(t)$ = state profile **bounds**
uL(t), $u^u(t)$ = control **profile bounds**

For the NLP formulation, we convert the differential equations to algebraic equations using collocation on finite elements (see Villadsen and Michelsen, 1978). Here the discretized collocation equations are evaluated at the shifted roots of an orthogonal polynomial, as shown in Figure L State and control profiles are represented in Lagrange form over an element i: $\zeta_i \leq t \leq \zeta_{i+1}$ and 0 £ x £ 1.

XK+I(0 = Lxij^jO) »j(t)-riLaj/t(ty,- in dementi (9)
k=o,j

u_K(0 = *L* uy 6j (t) 9j (t) = rifc.ij /*"*** in element i i = 1,... NE
j=i

Here k=lj denotes k * j. Also X£₊i(t) is a (K+1) th order piecewise polynomial and ujc(t) is a Kth order piecewise polynomial. (The difference in orders is due to the existence of the initial conditions for x(t), for each element i.)

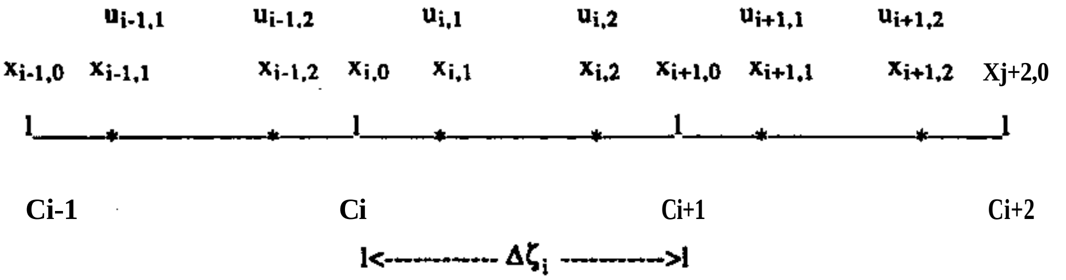

Figure 1: *Finite element collocation discretization for state and control profiles, and elements.*

Using K point, orthogonal collocation on finite elements (see Figure 1) and defining the polynomial basis functions so that they are normalized over each of the NE elements, one can write the collocation equations for the ODE system as follows.

A^i r(t_{ik}) = 2- Xy <t>j(Xk) -ACi F (x_{ik},u_{ik}) (10)
j = 0
i ~ 1,........,iNEr
k = 1........,K

where ^(xic) = -p- and is calculated offline. Note that t& = & + ACfo. Here the element lengths can also be included as decision variables in order to find possible points of discontinuity for the control profiles and to insure that the integration accuracy is within a specified tolerance. Additionally, we enforce the continuity of the states at element endpoints, i.e.:

$z^{i}_{K+1}(\zeta_i) = z^{i-1}_{K+1}(\zeta_i)$ or Ho-]L«l-iJ*|(t-i) i- 2,......,NE (11)

and the endpoints provide the initial conditions for the next element states. Given this approximation for the DAE model, problem (8) is now reformulated as follows:

Min ^ (x f, p) + X *Z*, WU G(Xij, uy , p,A£i)
i=1 j=1
x_{ij} , u_{ij} , p, $\Delta\zeta_i$ (12)
s.t. $\Delta\zeta_i\ r_{ij} = x_{K+1}(\tau_{ij})$ - A&F(Xy_fuy,p) =0
g (x_{ij} , u_{ij} , $\Delta\zeta_i$) ≤ 0 v

$g_f\ (X_f) \le 0$

$x_{10} - x_0 = 0$

XiO " x^{i-1}_{K+1} (Ci) = 0 $\quad i = 2,\ldots\ldots, NE$

Xf -· x^{NE} (CNE+I) = 0

$\le x_{ij}$ ^ x^{U}_{jj}

$u^{L}_{ij} \le u_{ij} \le u^{U}_{ij}$

$\Delta\zeta^{L}_{i} \le \Delta\zeta_{i} \le AC;^{U}$

$\overset{NE}{\underset{i=1}{I}}$- 4C --CTOUI

Note from the structure of the collocation equations that the control variables and element lengths directly determine the solution trajectories for the states. The linearized state equations are thus solved forward in time using the finite element structure and passing the information from element to element This allows us to exploit the sparsity of the ODE's and the collocation formulation. Once these trajectories have been computed, and the derivative information (sensitivity of states to control variables) is obtained, this information is chainruled in order to obtain the *reduced gradients* of the objective and contraint functions. We then construct the reduced QP subproblem in order to update the optimal control profile. This approach is especially efficient if the differential equations are *linear in the state variables* because die resulting method becomes a reduced gradient, feasible path approach, with the collocation equations solved at each optimization iteration (see Logsdon and Biegler, 1991a).

To illustrate this structured decomposition, let xj represent the interior states in element i; xj is determined in each element by collocation equations (10). In particular, from the collocation equations (hi = 0) we have:

hi (xio,Uij,Xij^j) = 0 $\quad {}^{AaB}\overset{\partial h\cdot}{\underset{oxy}{T}}$ ^ 'l . ≂ ^{1}--NE$_f$j = ^{1}—»^{K}

and, Ax!= - (A)"^{1}^ (xjo,Uij,Xij,A£j), a Newton step for the collocation equations (10). We further apply the linear continuity equations (11) to determine the initial conditions for the next finite element and continue the forward elimination of the collocation equations. This leads to the decomposition strategy for the Jacobian matrix, shown in Figure 2. Continuing die solution of the linearized collocation equations, we evaluate the final state variables as functions of the control variables, the element lengths and the initial state conditions, i.e.

xf = f(xo,ACi, ui,AC2. u_2, ..$\Delta\zeta$ ^u_f)

Note that the flow of information from element to element is passed forward through the continuity equations as illustrated in Figure 3. In addition, if we have inequality constraints that depend on state variables at final time or within some (or all) intermediate elements, i.e. (xc) at element c, then these can also be expressed by:

$x_c = f(x_0,\Delta\zeta_1, u_1,\Delta\zeta_2, u_2, \ldots.,\Delta\zeta_{c-1}, u_c)$

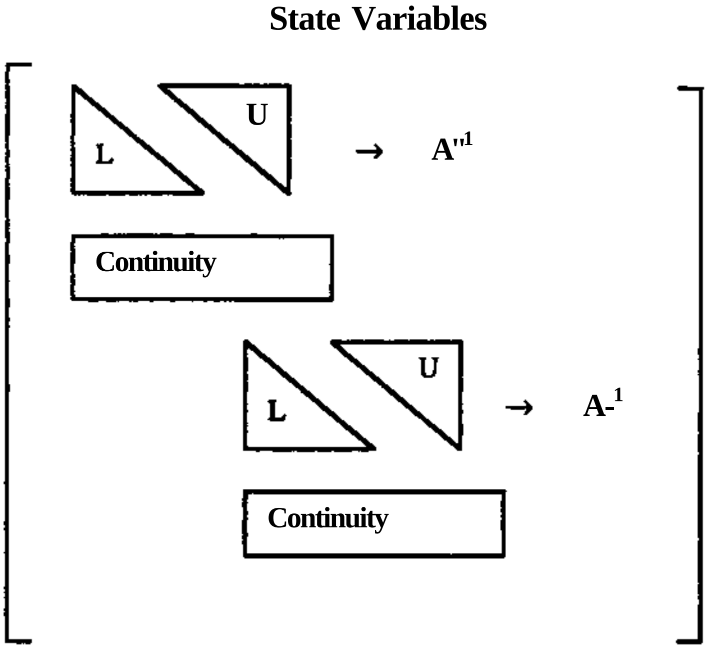

Figure 2: *Decomposition for Element to Element Solution Approach*

ui AC,

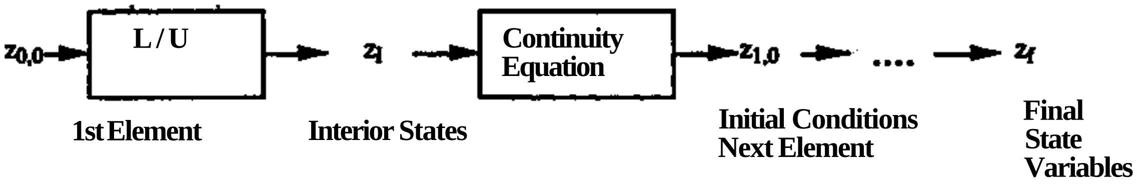

Figure 3: *ODE solver for state differential equations using collocation on finite elements witth information processed from element to clement.*

We then proceed to the next element, calculate the interior states and chainrule to obtain their sensitivity to the control variables from the previous element Note that the chainruling continues through the state variables at the end of element i, $\bar{x}_j$, starting from each control variable in every element j, up to element i.

$$\frac{\partial \bar{x}_i}{\partial u_j} = \frac{\partial \bar{x}_i}{\partial \bar{x}_{i-1}} \frac{\partial \bar{x}_{i-1}}{\partial \bar{x}_{i-2}} \cdots\cdots \frac{35}{\partial u_j} \qquad \begin{matrix} j \leq i \\ i=1,NE \\ J\text{-}U \end{matrix} \qquad (13)$$

This forward elimination and chainruling scheme acts as a linearized ODE solver and is exact if the state variables appear linearly in the DAE system. Once state variable vectors and their sensitivities are calculated, reduced gradients for the objective and ng constraint functions, g(xc), are constructed with respect to the jth control variable by the following relations:

$$\{Z^T\nabla\Phi\}_{j=} \frac{\partial z_i^*}{9UJ}\frac{\partial\Phi}{dzf} \qquad \{Z^T\nabla g_n)_r \frac{***}{9UJ}\frac{\pm}{3zc}\ n - 1.\ n, \quad (14)$$

Results of the gradient calculations are then transferred to a QP subproblem (15) which is solved in the control variable (and element length) space. A brief description of the algorithm tailored for DAE optimization is given next:

0. Choose the number of elements and the corresponding number of collocation points based on the likely index of the DAE system (see Logsdon and Biegler, 1989 for details). Initialize the control variables, state variables, and **element** lengths.

1. **For values** of the control variables **and element lengths at iteration k, and** initial conditions **for the state** variables, perform **the** following for each **element i (i = 1,...NE):**

   1.1 Using the **initial conditions of element i as starting guesses, (partially) solve the collocation equations (10) by (inexact) Newton algorithm to obtain an estimate of the interior states. (See Logsdon and Biegler, 1991b, for details on this inexact method.)**

   1.2 Calculate the derivatives with respect to this **element's** decision **variables.**

   1.3 Apply the **continuity equations (11), and solve for the next element's** initial condition.

   1.4 Chainrule the derivatives from **previous elements and update from equation (13).**

2. **Continue until an intermediate element is reached that influences an inequality (g(xc)), or until the last element is reached. Determine the reduced gradients for the objective and constraint functions according to equation (14).**

3. **Assemble the objective** and **all of the constraint function values and reduced gradients from the above steps. If Kuhn-Tucker conditions are satisfied, STOP. Otherwise solve the following quadratic program:**

$$\begin{aligned} &\mathrm{Min}_{Au}\ V{<}D^T Z\ Au + \frac{1}{2} A\, u^T (Z\ B^T Z)\ A\ U \\ &\text{s.t.}\quad g + Vg^T ZAu \quad £0 \end{aligned} \qquad (15)$$

to determine the search direction in u. Note that this QP contains all of the state and control variable inequality constraints. In addition, the reduced Hessian matrix, (Z^TBZ) is updated by the BFGS formula, and a line search is performed in order to determine the steplength for the decision variables.

5. **Return to step 1, with a new set of decision variables from (15).**

Note the similarity of this approach to the reduced SQP strategy outlined in the previous section. Again, a Newton step is first calculated for the dependent variables. Reduced gradients are then calculated for the independent variables and a quadratic program ((7) or (15)) determines the search direction for these variables.

To illustrate this structured approach, Logsdon and Biegler (1991b) consider a nonlinear batch reactor example (Ray, 1981) with temperature as the control variable. It is desired to maximize an intermediate product after a fixed reaction time. Here we consider the nonisothermal series reaction, A --> B --> C which is nonlinear in the rate equations. Letting ci and C2 represent the concentration of A and B, respectively, the optimal control problem becomes:

Max $c_2(1.0)$

dCi 2

dt

$\wedge$ = MT)c?- $k_2(T)c_2$

$ki(T) = A_{iO}\exp^{[}$ /RT$^{]}$ $i=1,2$

$c_1(0)=1.0$, $c_2(0) = 0$

$298 \leq T \leq 398$

Here two point collocation and six finite elements are adequate for the accuracy of the profiles and the discretized problem (12) has 54 variables. With a general purpose decomposition approach that does not take advantage of problem structure, we require 88 iterations for convergence (Logsdon and Biegler, 1989). With the structured approach and prescaling the reduced Hessian matrix, only 16 iterations and 33 CPU seconds (Vaxstation 3200) are required In both cases we started with an initial flat temperature profile of 300 K; the final control profile is shown in Figure 4. This approach has also been applied to much larger systems including optimization of reflux ratios for batch distillation columns, where nonlinear programs with over 5500 variables were solved and similar savings were observed (see Logsdon and Biegler, 1991b).

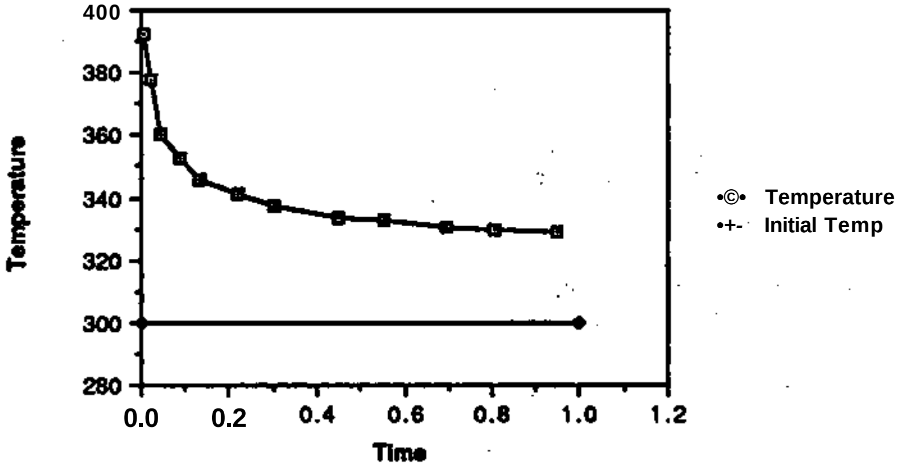

Figure 4: *Optimal Temperature Profile for Nonlinear Batch Reactor*

AN SQP METHOD FOR UNDERDETERMINED LEAST SQUARES PROBLEMS

In this section we consider a structured SQP algorithm for the parameter estimation problem represented by the following NLP formulation.

$$\min_{x} \Phi(x) \equiv \sum_{\mu=1}^{r} \phi_\mu = \frac{1}{2}\sum_{\mu=1}^{r} e_\mu^T W e_\mu \tag{16}$$

s. t.

$f_\mu(x_\mu,$ e> = o

$\mu = 1, 2, \ldots r,$

e <> e^L ^ e^U

where

$$x_\mu \in \Re^s, \quad \theta \in \Re^p$$

$$f_\mu : \Re^{s+p} \rightarrow \Re^m$$

$$e_{\mu k} \equiv x_{\mu k} - \bar{x}_{\mu k}$$

$$k = 1, 2, \ldots s$$

$e_{\mu k}$: the residual for k-th state variable at *\i-th* measurement

$x_{\mu k}$: the fined value for the k-th state variable at |i-th measurement

$\bar{x}_{\mu k}$: the measurement for k-th state variable at |i-th measurement

W : the positive semidefinite covariance matrix

This problem is often encountered in connection with reconciliation of process data, as well as estimation of model parameters from laboratory experiments. Here it is interesting to note two aspects of the problem which lead to more efficient, tailored SQP strategies. First the gradient of the objective function with respect to XJJ, vanishes with small residual values. From (6) it is clear that for problems with small residuals, the Lagrange multipliers also vanish at the solution. Consequently, an approximation to the Hessian of the Lagrange function often needs to consider only the second derivative terms in the objective function, which are available analytically. Tjoa and Bicglcr (1991a) exploited this property to develop a tailored quasi-Newton updating strategy for constrained least squares problems. In comparisons with general purpose SQP methods and MINOS, this approach was at least twice as fast

A second aspect of parameter estimation problems is that the regression model is often underdetermined. Here the number of independent variables is given by p + r (s - m) and this increases linearly with the number of data sets; a polynomial increase in effort is required by the NLP solver. To address this problem, Britt and Luccke (1973) and Dovi and Paladino (1989) develop efficient, special purpose algorithms to deal with underdetermined regression models. Here we show that application of a tailored SQP strategy also yields an efficient algorithm without sacrificing any of SQP's convergence properties. In particular, we note that the model parameters in (16) are the only complicating variables among data sets and the regression model can be uncoupled by adding the following dummy variables and equations: gp. = Cji - 0 = 0. With these additional variables and constraints, the QP subproblem for (16) at iteration k is given by:

$$\min_{X} \Psi = \sum_{\mu=1}^{r} \psi_\mu = \sum_{\mu=1}^{r} \left[(\nabla_{x_\mu} \phi_\mu)^T W \Delta x_\mu + \frac{1}{2} [\Delta x_\mu \quad \Delta\zeta_\mu]^T B_\mu \begin{bmatrix} \Delta x_\mu \\ \Delta\zeta_\mu \end{bmatrix} \right]$$

s. t.

$$(\nabla_{x_\mu} f_\mu^T) \Delta x_\mu + (\nabla_{\zeta_\mu} f_\mu^T) \Delta\zeta_\mu + f_\mu(x J\, g = 0 \qquad (17)$$

$$\Delta\zeta_\mu - A6 - 0$$

$$\mu = 1, 2, \ldots r$$

$$0^L \le; e + \Delta\theta \le; e^V,$$

where B^ the Hessian for each data set, includes W from die objective function and quasi-Newton approximations for the remaining terms from the Lagrange function. From the QP (17), the optimality conditions are decoupled for each data set and expressed as:

$$\begin{bmatrix} B_\mu & \nabla h_\mu \\ \nabla h_\mu^T & 0 \end{bmatrix} \begin{bmatrix} \Delta d_\mu \\ v_\mu \end{bmatrix} = - \begin{bmatrix} \nabla \phi_\mu \\ h^* J \end{bmatrix}, \qquad (18)$$

where

$$\nabla h_\mu = \begin{bmatrix} V_x f_a & 0 \\ V \text{rf cle} & I \end{bmatrix}, \quad \Delta d_\mu = \begin{bmatrix} \Delta x_\mu \\ \Delta \zeta_\mu \end{bmatrix}, \quad \text{and} \quad h_\mu = \begin{bmatrix} f_\mu \\ -A0 \end{bmatrix}.$$

To further exploit the structure in each data set, we apply a familiar decomposition strategy to (18). Here, again we define basis matrices ZJJ, and YJJ. such that

$$Z \qquad\qquad Z = 0.$$

and the structure of these matrices is identical for each data set. Substituting into (18) (as in eqn. (6)) leads to the following decomposed linear system for each data set:

$$\begin{bmatrix} Y_\mu^T B_\mu Y_\mu & Y_\mu^T B_\mu Z_\mu & \text{YjVly} \\ Z_\mu^T B_\mu Y_\mu & Z_\mu^T B_\mu Z_\mu & 0 \\ \nabla h_\mu^T Y_\mu & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{y_\mu} \\ d_{z_\mu} \\ v_\mu \end{bmatrix} = - \begin{bmatrix} Y_\mu^T \nabla \phi_\mu \\ Z_\mu^T \nabla \phi_\mu \\ h_\mu \end{bmatrix}. \quad (19)$$

Note the search directions dy and d_z for each data set can be calculated by using the third and second rows of (19), respectively. This calculation can be simplified further by exploiting the structure of the Hessian. Assuming that we have small residuals and have normalized the problem so that W = I, we note that the Lagrange multipliers in the approximation of B^ are also small and can be neglected. We thus introduce the following simplifications:

$$Z_\mu^T B_\mu Y_\mu = Z_\mu^T \begin{bmatrix} W & 0 \\ 0 & 0 \end{bmatrix} Y_\mu = 0 \qquad d_{y_\mu} = -(\nabla h_\mu^T Y_\mu)^{-1} h_\mu \qquad d_{z_\mu} = -(Z_\mu^T B_\mu Z_\mu)^{-1} Z_\mu^T \nabla \phi_\mu,$$

Note here that d_z^ is independent of AO and that, for any value of A6 and d_Zp., dy^ will satisfy the constraints in (17). We now construct the following reduced QP by summing the contributions in A6 from all of the data sets and substituting into (17). The resulting QP is only in the space of the p model parameters, 6:

$$\underset{6}{\text{Min}} \quad (\sum_{\mu=1}^{r} \alpha_\mu)\, A6 + 1A9^{1\Gamma}(\sum_{\mu=1}^{r} H_\mu)\, A6$$

$$\text{s. t.} \quad e^L \le e_k + \Delta\theta \le e^u,$$

and HM and Op, are constructed from the reduced gradients and the search directions dyp, and dzjj,. The advantages of solving this reduced QP are that we can also include parameter bounds and apply a line search strategy to enforce global convergence, as with any SQP method. Moreover, if the problem has zero residuals and the reduced Hessian of the Lagrange function for (16) is positive definite, this method is analogous to a Gauss-Newton (G-N) method and has a quadratic convergence rate. Otherwise, for large residual problems this approach has been generalized to include quasi-Newton updates for Z^TBZ and HJJ,. A detailed description of this Decoupled SQP (DSQP) algorithm is given in TjoaandBiegler (1991b).

To illustrate this approach, Tjoa and Biegler consider a simple regression model where the estimation problem can be made arbitrarily large by increasing die number of data sets. This model is taken from Rod and Hancil (1980) and has the following form:

$$f(x, 0) = x_2 - \theta_1 - \frac{1}{x_1 - \theta_2} = 0$$

Using the data generated in Tjoa and Biegler (1991b) we solve this problem for four cases where the number of data sets are 25, 50, 75, and 100. Here we note that for problems with few parameters and many data sets, the computational effort for DSQP increases only linearly and is largely characteristic of the model considered at each data set Consequently, this approach can be seen as a direct extension of the SQP decomposition strategy developed in the second section. Figure 5 briefly illustrates this property and compares this approach to the MINOS NLP solver. Note the polynomial increase in effort with problem size that is characteristic of all general-purpose NLP solvers, and the significant reduction in effort due to the structured decomposition in DSQP.

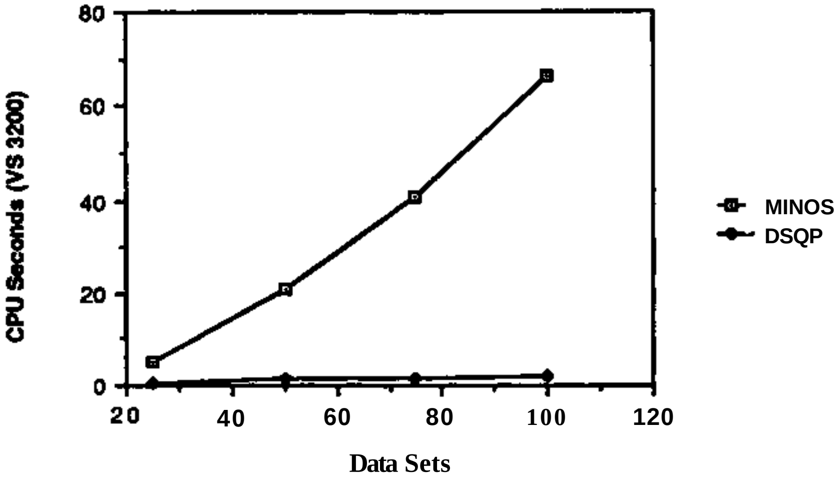

Figure 5: *Comparison ofMINOS and DSQP on Small Regression Example*

Finally, it should also be noted that this approach is not only restricted to parameter estimation problems. For example, it has straightforward extensions to multiperiod design problems that arise in the design of flexible flowsheets and heat exchanger networks. Moreover, the solution of the decoupled search directions in (19) is straightforward to execute in parallel and leads to the easy exploitation of coarse-grained parallel computer architectures.

SUMMARY AND CONCLUSIONS

The design of tailored Successive Quadratic Programming optimization strategies is outlined for three general applications in process design and analysis. Here we first consider the coupling of complex design models to a simultaneous optimization approach; we observe that reduced space SQP techniques not only lead to faster performance, but also allow the straightforward interface of Ncwton-Raphson model solvers to the optimization algorithm. This was briefly illustrated with a distillation example that interfaces to the Naphthali-Sandholm model in SEPSIM. Despite the strong coupling of the model equations and the optimization algorithm, this implementation allows broad classes of distillation problems to be specified and solved within the framework of the process simulation program.

This structured SQP approach was also extended to the optimization of differential-algebraic systems. Here the block lower triangular structure of collocation equations was exploited in the SQP decomposition step. After sketching the elements of this approach and drawing parallels to the previous section, we apply it to the optimization of a small batch reactor and compare it to an unstructured approach. Here the savings in effort are almost fourfold. Similar savings have also been observed on much larger problems involving the optimization of batch distillation units (Logsdon and Biegler, 1991b).

The last application deals with parameter estimation problems that have underdetermined process models. These problems have potentially many degrees of freedom, but also have a block diagonal substructure that allows for an efficient decomposition scheme. Here the algorithm is tailored to the structure of each data set; all of these have an identical structure. Consequently, the effort of this decoupled SQP (DSQP) approach increases only linearly with the number of data sets. To illustrate this strategy, the DSQP approach is applied to a simple regression model and compared to the MINOS algorithm. Here it is clear that, given their polynomial increase in effort with problem size, general purpose algorithms can require far more effort on these problems. Moreover, applications of this DSQP approach are under study for general multiperiod design problems.

The various decomposition strategies and illustrative examples attest to the flexibility of the SQP algorithm in exploiting the characteristics of process models. As shown above, additional development in tailored approaches for structured problem classes can lead to large savings in computational effort. An extremely convincing example of this is given in the last section. However, an open question remains as to how the model structure can be exploited easily by the casual user. Here flexible interfaces for the NLP solver and the process model require a fairly open data structure. With the application of powerful modelling tools with open architectures (such as ASCEND and SPEAKEASY) this task should, we hope, become straightforward and lead to more widespread development and application of model-tailored NLP solvers.

ACKNOWLEDGEMENTS

This work was partially supported by the Engineering Design Research Center, an NSF sponsored research center at Carnegie Mellon University. In addition, I am grateful to my colleagues, Jeff Logsdon, Claudia Schmid and Iauw-Bhieng Tjoa for numerous discussions, and for demonstrating this approach on the examples outlined above.

REFERENCES

Amarger R., L.T. Biegler and I.E. Grossmann, "REFORM: An Intelligent Interface for Formulating Optimization Problems," EDRC Report (1991)

Andersen, P. M., F. Genovese and J. Perregaard, "Manual for Steady State Simulator, SEPSIM," Institut for Kemiteknik, DTH, Lyngby, Denmark (1991)

Berna, T.. M.H. Locke and A.W. Westerberg, "A New Approach to Optimization of Chemical Processes, *AIChE /.*, 26, p. 37 (1980)**

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

Biegler, L.T., J. Nocedal and C. Schmid, "Reduced Hessian Strategies for Large Scale Nonlinear Programming," Working Paper (1991)

Britt, H. L, and R. H. Luecke, "The Estimation of Parameters in Nonlinear, Implicit Models, *Technometrics,* 15 (2), 233, 247 (1973).**

Chen, H-S, and MA Stadtherr, "A Simultaneous Modular Approach to Process Flowsheeting and Optimization," *AIChE /.*, 31,p. 1843(1985)

Dovi, V. C and O Paladino, "Fitting of Experimental Data to Implicit Models Using Constrained Variation Algorithm," *Computers and Chemical Engineering,* 13 (6), 731-735 (1989).

Gabay, D., "Reduced Quasi-Newton Methods with Feasibility Improvement for Nonlinearly Constrained Optimization," Math. Programming Study, 16, p. 18 (1982)

Han, S-P, "A Globally Convergent Method for Nonlinear Programming," J. Opt Theo. Applies, 22, p. 297 (1977)

Locke, M.H., R. Edahl and A. W. Westerberg, ^{M}An Improved Successive Quadratic Programming Optimization Algorithm for Engineering Design Problems," AICHE J, Vol. 29, no. 5 (1983)

Logsdon, J. S. and L. T. Biegler, "Decomposition Strategies for Large Scale Dynamic Optimization Problems," *Chem. Engr. ScL,* to appear (1991a)

Logsdon, J. S. and L. T. Biegler, "Reduced Oradient Strategies for Dynamic Optimization Problems," submitted for publication (1991b)

Logsdon, J. S. and L. T. Biegler, "On the Accurate Solution of Differential-Algebraic Optimization Problems, *I & EC Research,* 28, p. 1628 (1989)**

Lucia, A. **and** J. Xu, "Chemical Process Optimization Using Newton-Like Methods," 14, 2, p. 119 *Computers **and Chemical** Engineering* (1990)

Nickel, R.H. and J.W. Tolle, **"A** Sparse SQP Algorithm," *J. Opt. Theo. Applies..* 60, **p.** 453 (1989)

Noccdal, J. and M. Overton, "Projected Hessian Updating Algorithms for **Nonlinear Constrained Optimization,^{M} SIAM J. Num. Anal., Vol. 22, no. 5. p. 945** (1985)

Powell, M.J.D., "A Fast Algorithm for Nonlinear Constrained Optimization Calculations," *1977 **Dundee Conference on** Numerical Analysis,* (1977)

Ray, W.H., ***Advanced Process Control,*** **McGraw - Hill. New York (1981)**

Rod, V., and V. Hancil, 'Iterative Estimation of Model Parameters When Measurements of All Variables Are Subject to Errror," ***Computers and*** *Chemical Engineering,* 4, 33-38 (1980).

Schmid, C. and L.T. Biegler, "Acceleration of Reduced SQP Methods for Large-Scale Nonlinear Programming," presented at Annual AIChE Meeting, Los Angeles (1991)

Tjoa, I. B., and L. T. Biegler, "Simultaneous Solution and Optimization Strategies for Parameter Estimation of Differential -Algebraic Equation Systems," ***Ind. Eng. Chem. Res.*****, 30, 376 (1991a).**

Tjoa, I. B., and L. T. Biegler, "A Reduced SQP Strategy for Errors-in Variables Estimation," submitted for publication (1991b).

Vasantharajan, S., and L. T. Biegler, "Large-Scale Decomposition for Successive Quadratic Programming,"** ***Computers and Chemical Engineering,*** **12 (11), 1087-1101 (1988).**