On the Optimization of Differential-Algebraic Process Systems

Many chemical engineering problems require the optimization of systems of differential and algebraic equations. Here a method is presented based on finite-element collocation, which converts differential equations to algebraic residual equations with unknown coefficients. A nonlinear program is then formulated, with residuals incorporated as equality constraints and coefficients as decision variables. Also, adaptive knot placement is used to minimize the approximation error, with necessary and sufficient conditions for optimal knot placement incorporated as additional equality constraints in the nonlinear program. All equality constraints are then solved simultaneously with the optimization problem, thus requiring only a single solution of the approximated model. Finally, problems with discontinuous control profiles can be treated by introducing an extra level of elements (superelements) as decisions in the optimization problem. This approach is demonstrated on a simple optimal control problem as well as a reactor optimization problem with steep temperature profiles and state variable constraints.

J. E. Cuthrell, L. T. Biegler
Department of Chemical Engineering
Carnegie-Mellon University
Pittsburgh, PA 15217

Introduction

Optimization involving both differential and algebraic equation models represents one of the new frontiers facing optimization techniques. Currently, optimization of differential equations and algebraic equations can be done separately by various methods. Successive quadratic programming (SQP) (Han, 1977; Powell, 1977) and reduced gradient techniques (Murtagh and Saunders, 1978) are available for handling nonlinear algebraic problems, while variational calculus (Bryson and Ho, 1975) can be used for differential equation optimizations.

Up to this point, however, no general and accurate method is available for handling differential-algebraic optimization problems. Standard nonlinear programming (NLP) methods cannot be used without resorting to an often expensive numerical integration scheme. Furthermore, few options exist (Sargent and Sullivan, 1977), even in this case, for imposing inequality constraints on continuous state variable profiles and for handling discontinuities and/or singular control profiles. Methods based on optimal control theory, conversely, have difficulty handling even simple types of algebraic equations. Moreover, moderate-sized (say three or more state equations) nonlinear two-point boundary value problems, for example, composed of nonlinear ODE's can be difficult to solve using numerical integration. Furthermore, repeated numerical integration can make these optimizations very expensive. Additional complexities such as discontinuous/singular profiles, constraints, and other complex algebraic side conditions (Bryson and Ho, 1975) can make these problems almost impossible to solve using variational calculus.

In this paper we present a method that solves, under certain conditions, differential-algebraic optimization problems efficiently and accurately. Here the differential equations will be discretized using polynomial approximation and orthogonal collocation. The resulting algebraic equations are then written as part of an NLP that is solved with an SQP optimization technique. Accuracy of the approximation is guaranteed by developing a set of finite-element knot placement equations that represent sufficient conditions for error minimization. These equations also become part of the NLP. In addition to finite elements an extra level of elements, superelements, is introduced. Inclusion of these extra breakpoints allows the approximation of discontinuous/singular profiles.

A nonlinear program with ODE models

A differential-algebraic optimization problem (DAOP) that includes an ODE model (written here for the sake of conve-

Correspondence concerning this paper should be addressed to L.T. Biegler.

nience as an initial-value problem with ℓ between 0 and 1) can be posed as:

$$\begin{aligned}
\underset{x,U(\ell),Z(\ell)}{\text{Min}} \quad & \Phi[x, U(\ell), Z(\ell)] \\
\text{s.t.} \quad & c[x, U(\ell), Z(\ell)] = 0 \\
& g[x, U(\ell), Z(\ell)] \le 0 \\
& \dot{Z}(\ell) = F[x, U(\ell), Z(\ell), \ell] \quad \ell \in [0, 1] \\
& Z(0) = Z_0 \\
& x^L \le x \le x^U \\
& U^L \le U(\ell) \le U^U \\
& Z^L \le Z(\ell) \le Z^U \qquad \text{(DAOP)}
\end{aligned}$$

where Φ = an objective function, g, c = design constraint vectors, x = decision variable vector, $Z(\ell)$ = state profile vector (of dimension M), $U(\ell)$ = control profile, x^L, x^U = variable bounds, U^L, U^U = control profile bounds, Z^L, Z^U = state profile bounds.

The differential equations, as written, include a possible dependence on some scalar parameters (for example length of a reactor or pressure in a column). These have been chosen to include the decision variable vector x. The algebraic constraints and objective function in the DAOP are written as functionals, indicating a possible dependence upon the continuous control profile $U(\ell)$ and state profiles $Z(\ell)$.

As stated, a DAOP cannot be solved directly either by typical nonlinear programming techniques or by optimal control methods. In general, with an NLP technique one cannot optimize continuous profiles, nor is it possible to impose bounds and/or general constraints involving $Z(\ell)$ and $U(\ell)$. Optimal control methods, on the other hand, generally do not deal with algebraic constraints (such as c or g).

It is possible, under certain conditions, to handle a DAOP by using both an NLP optimization technique and a numerical integrator (Cuthrell and Biegler, 1985). This can be done in a straightforward manner only if no control profiles exist and no general constraints (including bounds) involving $Z(\ell)$ exist. Under these conditions, the DAOP can be solved with a two-step procedure where the optimizer chooses x in an outer loop, followed by numerical integration of the ODE model in the inner loop. This approach, however, requires the integration of sensitivity equations of the form

$$\frac{\partial \dot{Z}(\ell)}{\partial x} = \frac{\partial F[x, Z(\ell), \ell]}{\partial x} + \frac{\partial Z(\ell)}{\partial x} \frac{\partial F[x, Z(\ell), \ell]^T}{\partial Z(\ell)},$$

along with the ODE model, and can be expensive for both stiff problems and ODE's where the righthand sides are time-consuming to evaluate (e.g., for reactors in flowsheets). For some other special cases, such as algebraic equality constrained problems with no bounds or control profiles, a differential-algebraic equation (DAE) solver can be used with an NLP technique to solve the DAOP. However, this method still requires repeated solution of the DAE model (and sensitivity equations), and cannot be applied directly when general inequality constraints and/or boundary conditions are present.

A number of attempts have been made at solving various forms of DAOP. Lynn et al. (1970) addressed the simplest optimal control problem, i.e., minimize a function of terminal conditions subject to the ODE model. Here the necessary conditions were first developed using variational calculus, and then discretized. Later Lynn and Zahradnik (1970) applied the same idea to the problem of finding the optimal feedback control profile for a distributed system. In this case orthogonal (Chebyshev) polynomials and Galerkin's method were used to discretize a linear partial differential equation (PDE). This resulted in a standard linear-quadratic (LQ) control problem to which the earlier solution method was applied. Oh and Luus (1977) also discretized the variational necessary conditions for both the LQ problem and for a nonlinear model. Here orthogonal collocation at Legendre roots and power series polynomial approximations were used.

Neuman and Sen (1973a) also addressed the LQ problem, but also added state variable inequality constraints. This approach used cubic B-spline basis functions to approximate both the state and control profiles. The LQ problem is thus reduced to a quadratic program (QP) where the path constraints become just a system of linear constraints enforced at discrete points. The QP solution then yields approximations that are suboptimal, i.e., optimal with respect to the level of approximation. This method was then extended to a distributed-parameter system in Neuman and Sen (1973b) where Galerkin's method was used to discretize the PDE system, followed by application of the B-spline method to the resulting lumped system.

The LQ problem was also considered by Tsang et al. (1975). Here power series polynomial approximations were used and collocation at arbitrarily chosen points was performed.

Sargent and Sullivan (1977) considered a more general optimal control problem that included inequality constraints dependent upon both the states and control. In this approach the control profile was parameterized over variable time intervals, and the path constraints were transformed into constraints enforced at final time. The NLP, in the control parameters and time intervals, was then solved using a gradient-based method with the state and adjoint equations evaluated by numerical integration for a specified control profile.

Wong and Luus (1982) considered the problem of finding the optimal control profile for a linear parabolic PDE. Here the PDE was discretized using both global orthogonal collocation at Legendre roots and Lagrange interpolation polynomials. The resulting lumped system was then integrated repeatedly with the variables found by direct-search optimization.

Bapat and Heydweiller (1985) considered some optimal control problems, including reactor optimizations involving maximization of product subject to reaction kinetics and simple control bounds, and two controller design problems. Here Legendre polynomials were used to discretize both the state and control profiles. The resulting optimization problem in the control coefficients was then solved with a direct-search procedure with inequality constraints and/or variable bounds incorporated via a penalty function approach. Also, the nonlinear collocation equations were solved at each iteration with Newton's method. For problems containing a discontinuous control profile the time domain was divided into two elements each containing a state and control profile polynomial approximation. Here the location of the discontinuity was included as a variable in the optimization problem and found by the search procedure.

Each of the above approaches, while successful at addressing some of the problems associated with DAOP's cannot easily be extended to the general case. Those methods that discretized the problem and then applied a math programming technique (e.g.,

direct search or NLP) often used a relatively poor method of approximation, such as collocation at arbitrarily chosen points or global collocation, and usually did not address the question of approximation accuracy. Conversely, those methods using numerical integration can be expensive and are not easily applied to either boundary-value problems or problems with profile inequality constraints. As a result, we present a method in the next section that allows solution of general types of DAOP's.

An NLP Formulation for Differential-Algebraic Optimization Problems

This section focuses on two key ideas. First, by using orthogonal collocation the differential equations in a DAOP can be converted to a set of approximating algebraic equations. Second, the error of the approximation can be minimized by incorporating into the resulting NLP a finite-element method that adaptively chooses the locations of the finite-element knots.

Discretization of the ODE models

Recall the ODE model presented in the DAOP as:

$$\dot{Z}(\ell) = F[x, U(\ell), Z(\ell), \ell] \quad \ell \in [0, 1]$$
$$Z(0) = Z_0. \tag{1}$$

Next we make use of the two polynomials written in Lagrange form:

$$z_{K+1}(\ell) = \sum_{i=0}^{K} \mathscr{z}_i \phi_i(\ell) \quad \text{where} \quad \phi_i(\ell) = \prod_{k=0,i}^{K} \frac{(\ell - \ell_k)}{(\ell_i - \ell_k)} \tag{2}$$

$$u_K(\ell) = \sum_{i=1}^{K} \omega_i \psi_i(\ell) \quad \text{where} \quad \psi_i(\ell) = \prod_{k=1,i}^{K} \frac{(\ell - \ell_k)}{(\ell_i - \ell_k)}, \tag{3}$$

where $z_{K+1}(\ell)$ denotes a $(K+1)$th-order (degree $< K+1$) polynomial and $u_K(\ell)$ a Kth-order (degree $< K$) polynomial. [Note that $\phi_i(\ell)$ is a polynomial of degree K and $\psi_i(\ell)$ is a polynomial of degree $K-1$.] The difference in the orders is due to the existence of the initial condition for $Z(\ell)$. And, the notation $k = 0, i$ indicates that $k = 0, \ldots, i-1, i+1, \ldots, K$. The Lagrange form polynomial has the desirable property that [for $z_{K+1}(\ell)$ for example] $z_{K+1}(\ell_i) = \mathscr{z}_i$, which is due to the Lagrange condition $\phi_i(\ell_j) = \delta_{ij}$ (δ_{ij} = Kronecker delta).

Since for chemical engineering problems the states and controls represent quantities such as temperature or concentration, using Lagrange polynomials produces coefficients $\mathscr{z}_i$ and ω_i, which are physically meaningful quantities. This becomes useful when providing variable bounds, initializing a profile, or interpreting solution profiles. For other types of polynomial forms (e.g., power series polynomials or B-splines) the coefficients do not have these features.

Substitution of Eqs. 2 and 3 into Eq. 1 yields the residual equation:

$$R(\ell) = \sum_{i=0}^{K} \mathscr{z}_i \dot{\phi}_i(\ell) - F[x, u_K(\ell), z_{K+1}(\ell), \ell] \tag{4}$$

with $\mathscr{z}_0 = Z_0$, which still remains a function of time. Discretization of the residual equation is next done through use of one of the methods of weighted residuals (Villadsen and Michelsen, 1978; Finlayson, 1972). We choose the method of collocation, which requires

$$\int_0^1 R(\ell)\delta(\ell - \ell_i)\, d\ell = 0$$
$$i = 1, \ldots, K \quad (\delta = \text{Dirac delta}) \tag{5}$$

since this integral can be written simply as:

$$R(\ell_i) = \sum_{j=0}^{K} \mathscr{z}_j \dot{\phi}_j(\ell_i) - F[x, \omega_K(\ell_i), \mathscr{z}_{K+1}(\ell_i), \ell_i] = 0 \quad i = 1, \ldots, K$$

with $\mathscr{z}_0 = Z_0$.

With the Lagrange condition, the polynomials evaluated at discrete points reduce to the coefficient at that point, and thus

$$R(\ell_i) = \sum_{j=0}^{K} \mathscr{z}_j \dot{\phi}_j(\ell_i) - F(x, u_i, z_i, \ell_i), = 0 \quad i = 1, \ldots, K \tag{6}$$

with $\mathscr{z}_0 = Z_0$.

Similarly the polynomial approximations when substituted into the DAOP and evaluated at some ℓ_i become just the corresponding coefficient. Using the ODE model, Eq. 6, the DAOP becomes:

$$\begin{aligned}
\underset{x,\omega_i,\mathscr{z}_i}{\text{Min}} \quad & \Phi(x, \omega_i, \mathscr{z}_i) \\
\text{s.t.} \quad & g(x, \omega_i, \mathscr{z}_i) \le 0 \\
& c(x, \omega_i, \mathscr{z}_i) = 0 \\
& R(\ell_i) = \sum_{j=0}^{K} \mathscr{z}_j \dot{\phi}_j(\ell_i) \\
& \qquad - F(x, \omega_i, \mathscr{z}_i, \ell_i) = 0 \quad i = 1, \ldots, K \\
& \mathscr{z}_0 = Z_0 \\
& x^L \le x \le x^U \\
& U^L \le \omega_i \le U^U \\
& Z^L \le \mathscr{z}_i \le Z^U
\end{aligned} \tag{NLP1}$$

With NLP1 we can now solve very general differential-algebraic optimization problems once the points ℓ_i, $i = 1, \ldots, K$ have been chosen. An example of a flowsheet optimization problem solved with NLP1 can be found in Cuthrell and Biegler (1985). The locations of the points ℓ_i, $i = 1, \ldots, K$ are chosen to correspond to the shifted roots of an orthogonal Legendre polynomial of degree K (hence the term orthogonal collocation; Villadsen and Stewart, 1967). Orthogonal collocation can be equivalently thought of as a Galerkin procedure with the resulting integrals approximated by an optimal K-point quadrature. It turns out that the quadrature points for this approach correspond to roots of an orthogonal polynomial.

Essentially, the discretization of the ODE's is done over a set of points that can be visualized as shown in Figure 1 for $K = 3$. Here the initial condition specifies the first state coefficient, leaving MK residual equations in the $K(M+1)$ unknowns $\mathscr{z}_i, \omega_i$, $i = 1, \ldots, K$. The control profile coefficients are left as decision variables in the optimization, so the algebraic system of approximating equations has dimension $MK \times MK$.

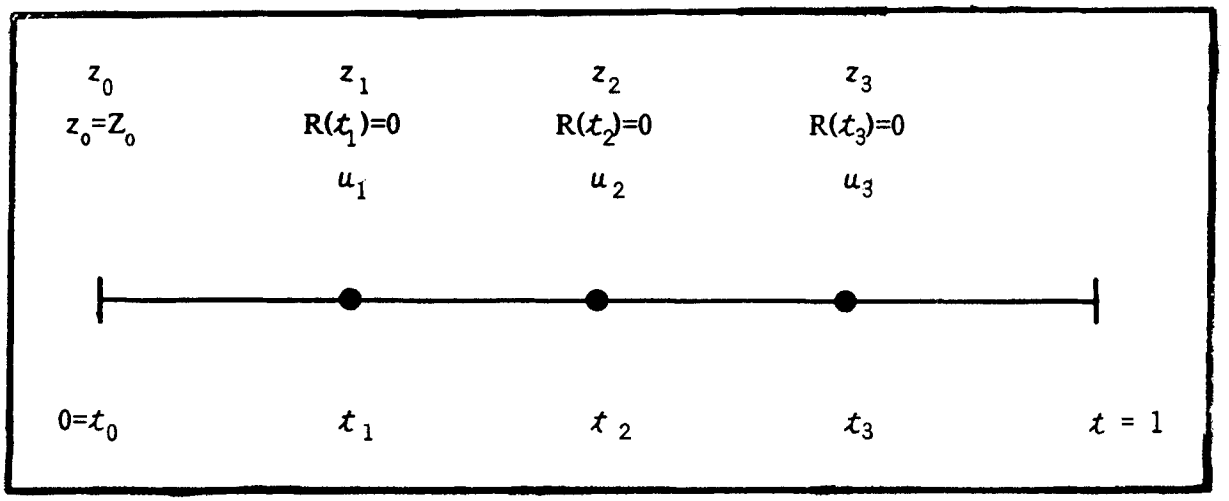

Figure 1. Global collocation.

Extension of orthogonal collocation to finite elements

The global collocation method discussed in the preceding section constructs the state approximation using $(K + 1)$th-order polynomials. For functions that are poorly behaved (i.e., rapidly changing in some small region) an accurate approximation using global collocation would require K to be very large. The steep region could then be approximated well at the expense of grossly overapproximating the rest of the function.

An alternative to global collocation uses piecewise polynomial approximations. Here a set of $(K + 1)$th-order state polynomials $z^i_{K+1}(\ell)$ and Kth-order control polynomials $u^i_K(\ell)$ are defined on finite elements. Each finite element $\Delta\alpha_i$ is bounded by two knots, α_i and α_{i+1}, with $\Delta\alpha_i \equiv \alpha_{i+1} - \alpha_i$. The distribution of elements now can be chosen so that the approximations are done both efficiently and accurately.

To begin a discussion of orthogonal collocation on finite elements, consider Figure 2, where $K = 2$, and $NE = 3$; NE is the number of finite elements. To preserve the orthogonal properties obtained with global collocation, the domain $\ell \in [0, 1]$ is mapped into each finite element through the formula (with $\alpha_1 = 0$, $\alpha_{NE+1} = 1$):

$$t = \alpha_i + \ell(\alpha_{i+1} - \alpha_i) \quad i = 1, \ldots, NE \text{ for } t \in [\alpha_i, \alpha_{i+1}].$$

And the locations of the orthogonal Legendre roots (with $\ell_0 \equiv 0$) are mapped to the points

$$t_{(i-1)(K+1)+j} = \alpha_i + \ell_j(\alpha_{i+1} - \alpha_i)$$
$$i = 1, \ldots, NE; \qquad j = 0, \ldots, K. \quad (7)$$

It is convenient at this point, in order to save a considerable amount of rewriting, to define the expression $(i - 1)(K + 1) + j$ as the new variable $[ij]$. The expression $[ij] \equiv (i - 1)(K + 1) + j$ is not to be confused with the commonly used double subscripts for matrices (e.g., A_{ij}, meaning the element in the ith row and jth column). Furthermore, the indices i and j can themselves take on other characters when the context requires it. For example, $[ik]$ becomes $(i - 1)(K + 1) + k$ for some i and k, and $[i1] = (i - 1)(K + 1) + 1$ for some i. For an equivalent

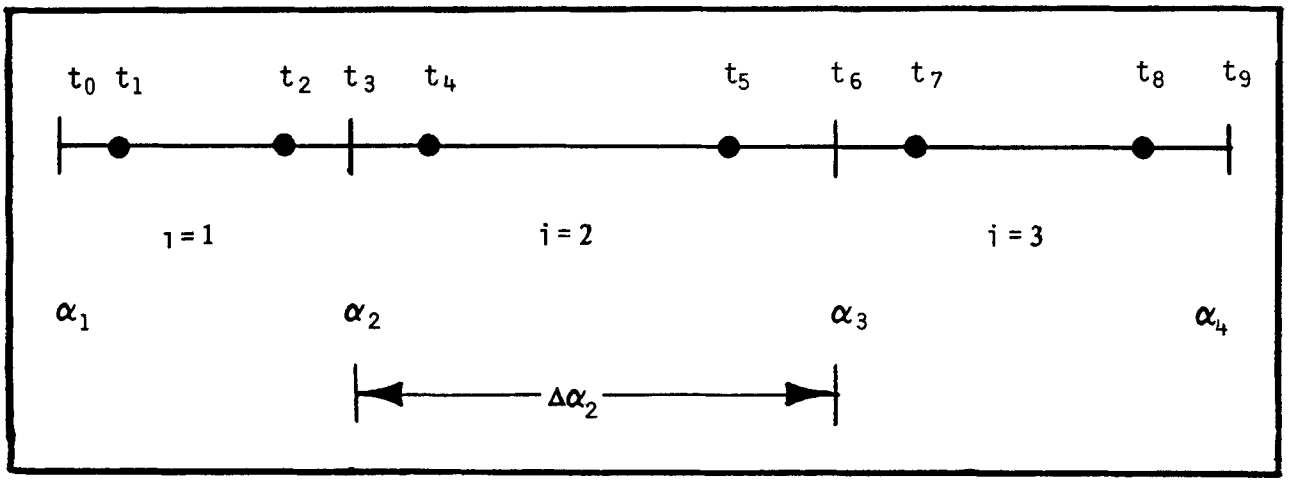

Figure 2. Finite-element collocation.

derivation of finite-element collocation that uses matrix notation, see Finlayson (1980). With this convention Eq. 7 becomes:

$$t_{[ij]} = \alpha_i + \ell_j(\alpha_{i+1} - \alpha_i)$$
$$i = 1, \ldots, NE; \qquad j = 0, \ldots, K. \quad (8)$$

The Lagrange polynomials can now be expressed as:

$$z^i_{K+1}(t) = \sum_{j=0}^{K} \mathscr{z}_{[ij]}\phi_{[ij]}(t) \quad \phi_{[ij]}(t) = \prod_{k=0,j}^{K} \frac{(t - t_{[ik]})}{(t_{[ij]} - t_{[ik]})}$$

$$u^i_K(t) = \sum_{j=1}^{K} \omega_{[ij]}\psi_{[ij]}(t) \quad \psi_{[ij]}(t) = \prod_{k=1,j}^{K} \frac{(t - t_{[ik]})}{(t_{[ij]} - t_{[ik]})} \quad (9)$$

for $i = 1, \ldots, NE$. For an equivalent representation of Eq. 9 without the previous notation change, see appendix A.

The discretized residuals can be written immediately from Eq. 6 as:

$$R(t_{[i\ell]}, \Delta\alpha_i) = \sum_{j=0}^{K} \mathscr{z}_{[ij]}\dot{\phi}_{[ij]}(t_{[i\ell]}) - F(x, \omega_{[i\ell]}, \mathscr{z}_{[i\ell]}, t_{[i\ell]}) = 0$$
$$i = 1, \ldots, NE; \qquad \ell = 1, \ldots, K \quad (10)$$

with $\mathscr{z}_{[10]} = Z_0$.

The calculation of the term $\dot{\phi}_{[ij]}(t_{[i\ell]})$ can be simplified by chain ruling derivatives to obtain:

$$\dot{\phi}_{[ij]}(t_{[i\ell]}) = \dot{\phi}_j(\ell_\ell)/\Delta\alpha_i$$
$$i = 1, \ldots, NE; j = 0, \ldots, K; \ell = 1, \ldots, K. \quad (11)$$

And thus Eq. 10 is more simply written

$$R(t_{[i\ell]}) = \sum_{j=0}^{K} \mathscr{z}_{[ij]} \frac{\dot{\phi}_j(\ell_\ell)}{\Delta\alpha_i} - F(x, \omega_{[i\ell]}, \mathscr{z}_{[i\ell]}, t_{[i\ell]})$$
$$= 0 \quad i = 1, \ldots, NE; \ell = 1, \ldots, K \quad (12)$$

with $\mathscr{z}_{[10]} = Z_0$.

In Eq. 12 the expression $\dot{\phi}_j(\ell_\ell)$ can be easily calculated off-line (Villadsen and Michelsen, 1978) since it depends only on the Legendre root locations. Now assuming for the moment that the variables x and $\omega_{[i\ell]}$ are fixed. Eq. 12 is composed of $M[NE(K) + 1]$ equations and $M[NE(K + 1)]$ state coefficients. To make the system well posed, an additional set of $M(NE - 1)$ equations is written to make the polynomials $z^i_{K+1}(t)$ continuous at the interior knots α_i, $i = 2, \ldots, NE$. This is done by enforcing

$$z^i_{K+1}(\alpha_i) = z^{i-1}_{K+1}(\alpha_i) \quad i = 2, \ldots, NE, \quad (13)$$

or equivalently

$$\mathscr{z}_{[i0]} = \sum_{j=0}^{K} \mathscr{z}_{[i-1,j]}\phi_j(\ell = 1) \quad i = 2, \ldots, NE. \quad (14)$$

These equations extrapolate the polynomial $z^{i-1}_{K+1}(t)$ to the end point of its element and provide an initial condition for the next element and polynomial $z^i_{K+1}(t)$. Each overall approximation to the state profile is therefore a continuous and piecewise polynomial function of order $K + 1$. Stated mathematically, $z_{K+1}(t) \in \mathbf{P}_{K+1} \cap \mathbf{C}[a, b]$ where $\mathbf{P}_{K+1}$ denotes the set of all polynomials of order $K + 1$ and $\mathbf{C}[a, b]$ the set of continuous functions. A number of authors construct differentiable and piecewise polynomial approximations, from $z_{K+1}(t) \in \mathbf{P}_{K+1} \cap \mathbf{C}^1[a, b]$, to higher order ODE or PDE systems (Finlayson, 1980; Gardini et al., 1985). However, continuous approximations are sufficient for our case, particularly since our examples have nondifferentiable solution profiles.

Including the ODE model, discretized on finite elements, and the continuity conditions at the knots, the NLP formulation becomes:

$$\begin{aligned}
\underset{x,\omega_{[i\ell]},\mathscr{z}_{[i\ell]}}{\text{Min}} \quad & \Phi(x, \omega_{[i\ell]}, \mathscr{z}_{[i\ell]}) \\
\text{s.t.} \quad & g(x, \omega_{[i\ell]}, \mathscr{z}_{[i\ell]}) \le 0 \\
& c(x, \omega_{[i\ell]}, \mathscr{z}_{[i\ell]}) = 0 \\
& R(t_{[i\ell]}) = 0 \quad i = 1, \ldots, NE; \quad \ell = 1, \ldots, K \\
& \mathscr{z}_{[10]} = Z_0 \\
& \mathscr{z}_{[i0]} = \sum_{j=0}^{K} \mathscr{z}_{[i-1,j]}\phi_j(\ell = 1) \quad i = 2, \ldots, NE \\
& x^L \le x \le x^U \\
& U^L \le \omega_{[i\ell]} \le U^U \\
& Z^L \le \mathscr{z}_{[i\ell]} \le Z^U
\end{aligned} \quad \text{(NLP2)}$$

With α_i, $i = 2, \ldots, NE$ fixed, NLP2 can be used to solve most types of differential-algebraic optimization problems. The importance of accuracy of the approximation however has not been discussed. This we take up in the next section, where a set of conditions is developed that guarantees minimization of the approximation error at the solution of NLP2.

Development of an error minimization strategy

In this section we present a set of error minimization or knot placement equations. These equations involve the state profile approximations $z^i_{K+1}(t)$, $i = 1, \ldots, NE$, and the knot locations α_i, $i = 1, \ldots, NE + 1$. To make the presentation of these ideas simpler we shall treat the M vector $Z(t)$ as a scalar for the next three subsections. In the fourth subsection this restriction will be relaxed to include multiple states ($M > 1$). At the NLP solution these knot placement equations require the interior knots to be at locations for which the approximation error is minimized. The entire development of the error minimization equations including the following theorem is presented in Cuthrell (1986) and will only be outlined here. We begin this section by stating a theoretical bound on the approximation error in Theorem 1 and develop from this and some assumptions the knot placement equations.

Theorem 1.

$$E_{K+1}\{Z(t); [\alpha_i, \alpha_{i+1}]\} \le C\Delta\alpha_i^{K+1} \| Z^{(K+1)}(t) \|_i \quad (15)$$

where:

- $E_{K+1}\{Z(t); [\alpha_i, \alpha_{i+1}]\}$ represents the local error between an optimal approximating polynomial of $(K + 1)$th-order and the function $Z(t)$ over some region $[\alpha_i, \alpha_{i+1}]$.
- C is a calculable (but physically of little use) constant dependent only upon K.

• $\Delta\alpha_i^{K+1}$ is the $(K+1)$th-power of the ith finite element length.

• $\|Z^{(K+1)}(t)\|_i$ is the max-norm of the $(K+1)$th derivative of the function $Z(t)$ in the interval $[\alpha_i, \alpha_{i+1}]$.

An Error Minimization Problem. To minimize the approximation error we formulate the following NLP.

$$\begin{aligned} \min_{\alpha_i} \max_j \ & \Delta\alpha_j \|Z^{(K+1)}(t)\|_j^{1/(K+1)} \quad i = 2, \ldots, NE \\ & \qquad\qquad\qquad\qquad\quad j = 1, \ldots, NE \\ \text{s.t.} \quad & \Delta\alpha_i \geq \epsilon \quad i = 1, \ldots, NE \\ \text{with} \quad & \alpha_1 = 0, \alpha_{NE+1} = 1 \\ & \epsilon = \text{a small positive constant} \end{aligned} \tag{NLP3}$$

Minimization of the approximation error can now be done, along with solving the discrete DAOP, by solving NLP3 in addition to NLP2. Specifically, we minimize $\Phi(x, \omega_{[it]}, \mathcal{F}_{[it]})$ in NLP2 over $x, \omega_{[it]}, \mathcal{F}_{[it]}$, subject to the indicated constraints and subject to NLP3 being solved. The resulting optimization problem can be greatly simplified, provided $\Delta\alpha_i > \epsilon$ holds, by noting that the necessary optimality conditions of NLP3 are:

$$\Delta\alpha_i \|Z^{(K+1)}(t)\|_i^{1/(K+1)} = \text{constant} \quad i = 1, \ldots, NE, \tag{16}$$

(Cuthrell, 1986). Furthermore, it can be shown (Fiacco, 1976) that Eq. 16 also represents the sufficient optimality conditions for NLP3.

Equation 16 can now be used as a basis for developing a set of working knot placement equations that can be included in NLP2 as equality constraints.

Development of Knot Placement Equations. Numerous strategies exist in the approximation theory literature for choosing a "good" distribution of knots when using finite elements. An excellent review, with an accompanying numerical comparison, is given in Russell and Christiansen (1978). The method presented here is a slightly modified version of one developed by deBoor (1978) and based on Eq. 16. It is not a fully rigorous method, involves convergence of highly nonlinear equations, and can fail under circumstances that will be outlined later. However, both papers report the method to be quite effective on a number of examples, and our results support this conclusion.

Recall that Eq. 15 represents a local approximation error. A much more detailed analysis of polynomial approximation using collocation at Legendre roots (deBoor, 1974; deBoor and Swartz, 1973 results in the addition of the global term $o(\Delta\alpha_i^{K+1})$. This term, however, is difficult to quantify and vanishes for small $\Delta\alpha_i$. We therefore assume it is negligible, and instead focus our efforts on the local term. The notion of equidistributing the local term, that is, distributing the local term equally over the entire range $[a, b]$ forms the basis for most of the knot placement methods we are aware of, even many not based on Eq. 16. Our task therefore is to develop a simple strategy that equidistributes the error bound given by Eq. 16.

Two difficulties immediately arise from Eq. 16. First, the infinity norm must be calculated, and second, we also need derivatives of the function $Z(t)$, which itself is unavailable since it is the true solution of the differential equations.

Notice first that the solution to Eq. 16, in terms of α_i, is asymptotically equivalent (i.e., as $\Delta\alpha_i \to 0$) to the solution of:

$$\int_{\alpha_i}^{\alpha_{i+1}} |Z^{(K+1)}(t)|^{1/(K+1)}\, dt = \int_{\alpha_{i+1}}^{\alpha_{i+2}} |Z^{(K+1)}(t)|^{1/(K+1)}\, dt \qquad i = 1, \ldots, NE-1. \tag{17}$$

For a proof of this equivalence see Russell and Christiansen (1978) and Pereyra and Sewell (1975). Evaluation of Eq. 17 is still not possible since the integrand is unknown. However, if we can find some constant $s_i \sim |Z^{(K+1)}(t)|_i$ then Eq. 17 reduces to:

$$s_i^{1/(K+1)} \Delta\alpha_i = \text{constant} \quad i = 1, \ldots, NE. \tag{18}$$

The problem of solving Eq. 17 thus becomes one of finding s_i and showing that it approximates $|Z^{(K+1)}(t)|_i$.

To begin the derivation of s_i, consider Figure 3, where for illustration purposes $NE = 3$ will be used. Here we use the shorthand $\Theta_{i+1/2} \equiv Z^{(K)}(\alpha_{i+1/2})$, $i = 1, 2, 3$, and also $\alpha_{i+1/2} = \frac{1}{2}(\alpha_i + \alpha_{i+1})$. The $\tilde{s}_i$'s are now divided difference approximations to $Z^{(K+1)}$, i.e.,

$$\tilde{s}_1 \equiv \frac{\Theta_{5/2} - \Theta_{3/2}}{\alpha_{5/2} - \alpha_{3/2}}$$

$$\tilde{s}_2 \equiv \frac{\Theta_{7/2} - \Theta_{5/2}}{\alpha_{7/2} - \alpha_{5/2}}.$$

Observe further that $\lim_{\alpha_{5/2} \to \alpha_{3/2}} \tilde{s}_1 = Z^{(K+1)}(\alpha^{3/2})$ and $\lim_{\alpha_{7/2} \to \alpha_{5/2}} \tilde{s}_2 = Z^{(K+1)}(\alpha_{5/2})$. So the divided differences approximations approach the desired derivative in the limit as $\Delta\alpha \to 0$.

Note also that $\lim_{\Delta\alpha \to 0} \tilde{s}_i = Z^{(K+1)}(\alpha_{i+1/2})$ implies $\lim_{\Delta\alpha \to 0} |\tilde{s}_i| = |Z^{(K+1)}(\alpha_{i+1/2})|$.

For $NE = 3$. Eq. 18 becomes $s_1^{1/(K+1)}\Delta\alpha_1 = s_2^{1/(K+1)}\Delta\alpha_2 = s_3^{1/(K+1)}\Delta\alpha_3$, and we can obtain s_i, from $\tilde{s}_i$ using the following heuristic:

$$\begin{aligned} s_1 &= |\tilde{s}_1| \\ s_i &= \tfrac{1}{2}|\tilde{s}_i| + |\tilde{s}_{i+1}| \quad i = 2, \ldots, NE-1 \\ s_{NE} &= |\tilde{s}_{NE}| \end{aligned}$$

For the case $NE = 3$, and using $\Delta\Theta_{i+1/2} = \Theta_{i+3/2} - \Theta_{i+1/2}$, this yields

$$\begin{aligned} s_1 &= \frac{2|\Delta\Theta_{3/2}|}{\alpha_3 - \alpha_1} \\ s_2 &= \frac{|\Delta\Theta_{3/2}|}{\alpha_3 - \alpha_1} + \frac{|\Delta\Theta_{5/2}|}{\alpha_4 - \alpha_2} \\ s_3 &= \frac{2|\Delta\Theta_{5/2}|}{\alpha_4 - \alpha_2}. \end{aligned} \tag{19}$$

Now it is easy to show that the limits of s_i, $i = 1, 2, 3$ as $\Delta\alpha \to 0$ all converge to the desired derivative $Z^{(K+1)}$. However since $Z(t)$, and all of its derivatives, are unavailable we assume further that $[z_{K+1}^{(K)}]^i \sim Z^{(K)}(t)$ over $t \in [\alpha_i, \alpha_{i+1}]$ $\forall$ i.

Making this substitution and rewriting Eq. 19 for the general case leads to the following equations (first derived in a different

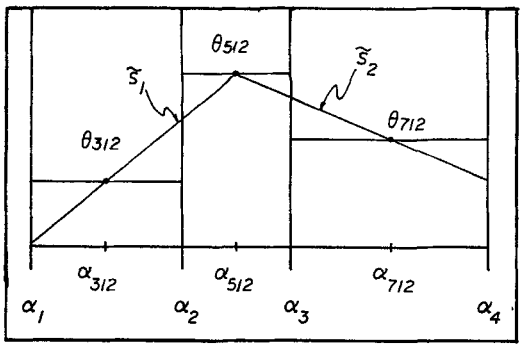

Figure 3. Calculation of slopes s_i.

way by deBoor, 1978).

$$s_i = \begin{cases} \dfrac{2|\Delta\Theta(\alpha_{3/2})|}{\alpha_3 - \alpha_1} & \text{on } (\alpha_1, \alpha_2] \\ \dfrac{|\Delta\Theta(\alpha_{i-1/2})|}{\alpha_{i+1} - \alpha_{i-1}} + \dfrac{|\Delta\Theta(\alpha_{i+1/2})|}{\alpha_{i+2} - \alpha_i} & \text{on } [\alpha_i, \alpha_{i+1}];\ i = 2, \ldots, NE-1 \\ \dfrac{2|\Delta\Theta(\alpha_{NE-1/2})|}{\alpha_{NE+1} - \alpha_{NE-1}} & \text{on } [\alpha_{NE}, \alpha_{NE+1}] \end{cases} \quad (20)$$

where

$$\Theta(\alpha_{i+1/2}) = [z^{(K)}_{K+1}]^i \quad i = 1, \ldots, NE$$

$$= \text{the highest nonzero derivative of } z^i_{K+1}(t) \text{ in } \Delta\alpha_i$$

Equation 20 can now be used with the following Eq. 21 as a set of knot placement equations that can be added directly to NLP2 as equality constraints. The knot placement equations written below include the explicit functional dependence on all the appropriate variables for the purpose of completeness. From this point forward, however, the shorthand $h_i = 0$, $i = 1, \ldots, NE - 1$ will be used. (Note that a minimum of three finite elements is always needed or effectively no knot placement can occur.)

$$h_1(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \mathcal{Y}_{[1j]}, \mathcal{Y}_{[2j]}, \mathcal{Y}_{[3j]}) = s_1^{1/(K+1)}\Delta\alpha_1 - s_2^{1/(K+1)}\Delta\alpha_2 = 0$$

$$h_i(\alpha_{i-1}, \ldots, \alpha_{i+3}, \mathcal{Y}_{[i-1j]}, \mathcal{Y}_{[ij]}, \mathcal{Y}_{[i+1j]}, \mathcal{Y}_{[i+2j]}] = s_i^{1/(K+1)}\Delta\alpha_i - s_{i+1}^{1/(K+1)}\Delta\alpha_{i+1} = 0 \quad i = 2, \ldots, NE-2$$

$$h_{NE-1}(\alpha_{NE-2}, \alpha_{NE-1}, \alpha_{NE}, \alpha_{NE+1}, \mathcal{Y}_{[NE-1j]}, \mathcal{Y}_{[NEj]}, \mathcal{Y}_{[NE+1j]}) = s_{NE-1}^{1/(K+1)}\Delta\alpha_{NE-1} - s_{NE}^{1/(K+1)}\Delta\alpha_{NE} = 0 \quad (21)$$

for $j = 0, \ldots, K$ (except $NE = 1$ and $j = 0$; initial condition).

Knot Placement Equations with Multiple State Profiles. A modified form of Eq. 21 can be used if it is desirable to include the influence of the errors from multiple profiles. This results in the knot locations being at positions for which some overall error is equidistributed.

Recall that the state profile vector $Z(t)$ has dimension M, which will now be treated explicitly. Approximations of these states on finite elements then yields the polynomials $[z^i_{K+1}(t)]_m$ for $i = 1, \ldots, NE$, $m = 1, \ldots, M$. Equation 20 can then be constructed for each state profile with the Eq. 21 becoming:

$$h_i = \left[\sum_{m=1}^{M} (s_i^{2/(K+1)})_m\right]^{1/2}\Delta\alpha_i - \left[\sum_{m=1}^{M} (s_{i+1}^{2/(K+1)})_m\right]^{1/2}\Delta\alpha_{i+1} = 0 \quad (22)$$

for $i = 1, \ldots, NE - 1$.

Equation 22 incorporates the errors, in each element i, associated with all states $m = 1, \ldots, M$, into the knot placement equations through a 2-norm. (The 2-norm of $s_i^{1/(K+1)}$ is required over the theoretically more desirable max-norm because of differentiability considerations.)

An NLP method for optimizing differential-algebraic systems

This section simply states the NLP formulation that can accurately optimize differential-algebraic systems of equations. Included are the residual approximations and continuity relations at the knots, as well as the knot placement equations.

$$\begin{aligned} \underset{x, \omega_{[i\ell]}, \mathcal{Y}_{[i\ell]}, \alpha_i}{\text{Min}} \quad & \Phi(x, \omega_{[i\ell]}, \mathcal{Y}_{[i\ell]}) \\ \text{s.t.} \quad & g(x, \omega_{[i\ell]}, \mathcal{Y}_{[i\ell]}) \le 0 \\ & c(x, \omega_{[i\ell]}, \mathcal{Y}_{[i\ell]}) = 0 \\ & R(t_{[i\ell]}, \Delta\alpha_i) = 0 \quad i = 1, \ldots, NE; \quad \ell = 1, \ldots, K \\ & \mathcal{Y}_{[10]} = Z_0 \\ & \mathcal{Y}_{[i0]} = \sum_{j=0}^{K} \mathcal{Y}_{[i-1j]}\phi_j(\ell = 1) \quad i = 2, \ldots, NE \\ & h_i = 0 \quad i = 1, \ldots, NE-1 \\ & \alpha_{i+1} \ge \alpha_i + \epsilon \quad i = 1, \ldots, NE \\ & x^L \le x \le x^U \\ & U^L \le \omega_{[i\ell]} \le U^U \\ & Z^L \le \mathcal{Y}_{[i\ell]} \le Z^U \end{aligned} \quad \text{(NLP4)}$$

The procedure used in NLP4 to position the knots differs from previous ODE solving methods that use orthogonal collocation on finite elements. deBoor (1978) selects the knots in an outer loop and then solves the collocation equations. Repositioning of the knots is then done to obtain a better approximation, with the procedure terminating when knot movement is very small. Ascher et al. (1979) use a similar procedure in the FORTRAN package COLSYS. Here an extrapolation technique is used to bound the approximation error, which requires repeated solutions of the approximated problem. Based on this error the knots are either repositioned or new knots are introduced (which increases the size of the problem) by bisecting all intervals.

Our method differs from those of deBoor and Ascher et al., in that we solve the knot placement and collocation equations simultaneously. Furthermore, we use the more desirable Lagrange basis polynomials, as opposed to the B-splines used in each above reference. The simultaneous procedure we use has distinct advantages that will be pointed out in the later section on optimization of a reactor problem. The ability to change the size of the problem by introducing new knots, however, is not a feature of NLP4.

To introduce the next concept, however, we first discuss a type of problem that NLP4 cannot solve. Here a number of changes are necessary for the method to handle more general types of problems.

Extension of the Method to Problems with Discontinuities

Example problem P1: a failure of NLP4

A number of problems that occur frequently in the area of optimal control are those which have discontinuous and/or singular arc control profiles. Consider the following example problem:

$$\begin{aligned} &\underset{U(t),T}{\text{Min}} \quad T \\ &\text{s.t.} \quad \dot{Z}_1(t) = Z_2(t) \qquad Z_1(0) = 0 \quad Z_1(T) = 300 \\ &\qquad \dot{Z}_2(t) = U(t) \qquad Z_2(0) = 0 \quad Z_2(T) = 0 \\ &\qquad -2 \le U(t) \le 1 \end{aligned} \tag{P1}$$

which represents the problem of finding the minimum time to cover 300 distance units, starting and stopping at rest, subject to acceleration limits of -2 and 1. This problem is linear in $U(t)$ and has an analytic solution with a bang-bang control profile. The optimal value of T is 30 and the point of discontinuity (or switching time t_{sw}) occurs at 20. The optimal state profiles are drawn as solid lines in Figure 4 with the complete analytical solutions given below

$$\begin{aligned} \text{For } t \in [0, 20): \quad & U(t) = 1 \\ & Z_1(t) = \tfrac{1}{2} t^2 \\ & Z_2(t) = t \\ \text{For } t \in (20, 30]: \quad & U(t) = -2 \\ & Z_1(t) = -t^2 + 60t - 600 \\ & Z_2(t) = 60 - 2t \end{aligned}$$

Note that the optimal control profile is a piecewise constant function and the state profiles are simply piecewise polynomials of degree 2 or less. Problem P1 was solved using NLP4, with both two and three finite elements, and with $K = 2$. Since $Z_1(t)$ is a quadratic at the optimal solution, both approximations were constructed using quadratics. Given that the optimal $u_K(t)$ is found accurately, this results in the approximations for both $Z_1(t)$ and $Z_2(t)$ being exact. The solutions obtained using NLP4 are presented in Table 1 and Figures 4 and 5. For the state profiles the analytical solutions are indicated with solid lines and the approximations with broken lines. For the control profile only the approximate solutions are presented, as solid lines.

Clearly, NLP4 fails to solve P1, with neither the correct T, t_{sw} nor the profiles being found accurately. The most obvious reason is that the bang-bang control profile is approximated poorly in both cases (although it appears to have improved for $NE = 3$). It is instructive at this point to investigate some factors that did not contribute to this failure. First, the order of the polynomial

Table 1. Results for Problem P1 Using NLP4

	$NE = 2$	$NE = 3$
$T(=\alpha_{NE+1})$	30.5154	30.2933
α_i $i = 2, \ldots, NE$	15.2577	12.9002, 22,2971

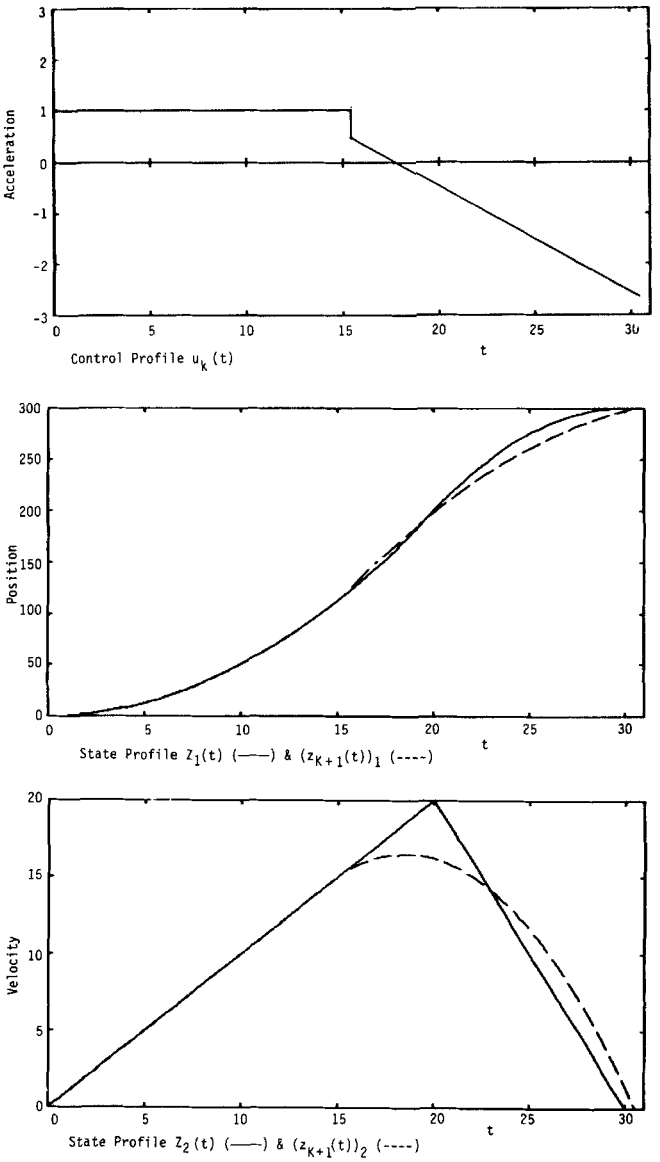

Figure 4. Control and state profiles, *NE* = 2.

approximation is sufficient (quadratics were used). Second, the number of finite elements used is also high enough (compared to the analytical solution). Here the two-finite-element case, at first glance, should have been acceptable since there is only one point of discontinuity. Third, the degree of polynomial continuity enforced is adequate. Recall that states are required only to be continuous (and here both analytical state profiles are nondifferentiable at t_{sw}), and that control profile continuity at the knots is not enforced.

It is clear from these observations that the solution to the knot placement equations has precluded the locations of the knots being at positions of control profile discontinuity. Therefore an additional set of elements (or knots) is needed that is independent of the knot placement equations. This motivates our discussion of superelements, presented next.

Superelements: problem P1 solved

We now define a new level of elements, termed superelements, defined by $\Delta\xi_{ise}$, $ise = 1, \ldots, NSE$ with the help of Fig-

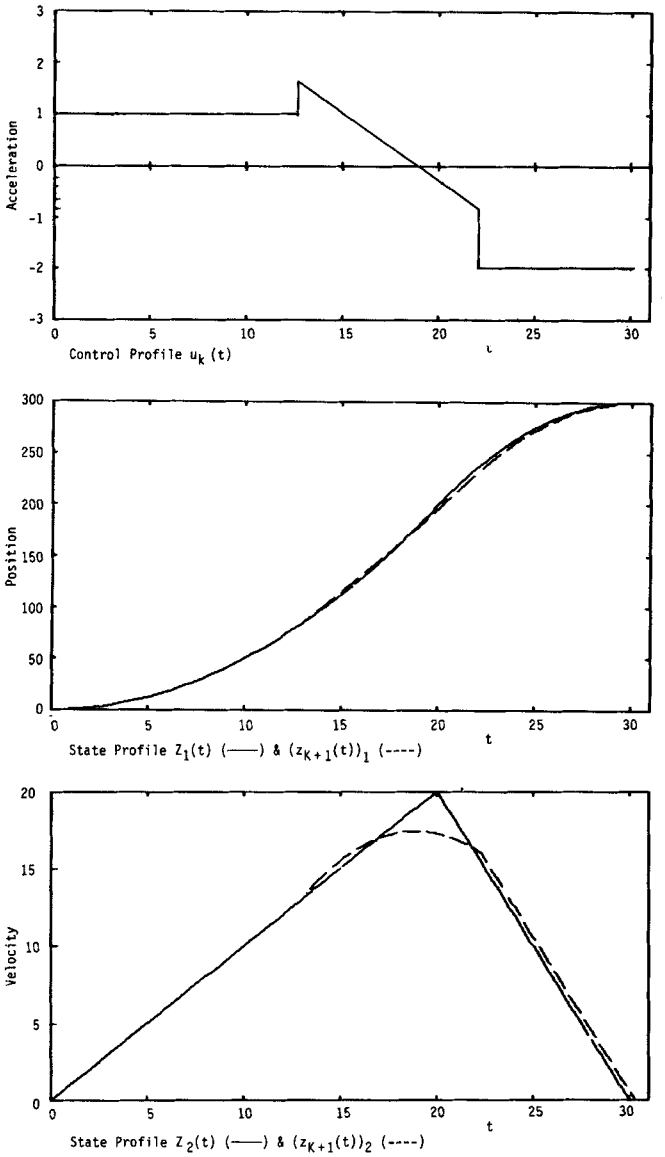

Figure 5. Control and state profiles, *NE* = 3.

ure 6 (for $K = 2, NE = 3, NSE = 2$; NSE is the number of superelements). A superelement, $\Delta\xi_{ise}$, is composed of a set of NE finite elements and is bounded by two breakpoints ξ_{ise}, ξ_{ise+1} with $\Delta\xi_{ise} \equiv \xi_{ise+1} - \xi_{ise}$. For the remainder of this paper the term breakpoint will refer to a superelement boundary and is to be distinguished from the term knot (which refers to finite-element boundaries). In this strategy breakpoints are left completely free, to be chosen to correspond to the optimal locations of control profile discontinuity. Within each superelement the knot placement equations, Eq. 22, are applied to obtain accurate state approximations. Also, with the addition of this extra level of elements, state profile continuity must be enforced at all interior breakpoints. The control profiles however can be discontinuous at the breakpoints.

This modification makes it possible to solve problems that have both control profile discontinuities and state profiles that are hard to approximate. The ability to do this, however, comes at the expense of perhaps having to solve a large optimization problem. Here the number of equations and variables representing the approximation grows large as NE is increased, and almost doubles as NSE is increased from 1 to 2. Thus there is the potential for dealing with very large (>100 variables and equations) NLP's.

To demonstrate the need to use superelements, consider again problem P1. Recall that quadratic ($K = 2$) approximations of the states will be exact if the control profile is found accurately. The use of finite elements for this problem is therefore unnecessary; their only function is to ensure accuracy of the state profiles. As a result P1 was resolved using $K = 2$, $NSE = 2$ and 3, and no finite elements. The results are presented in Table 2 (with $\xi_1 = 0$ and noting that $\xi_{NSE+1} = T$).

The results point out clearly that given enough flexibility, the method can solve problems that have discontinuous profiles. Here the correct T and t_{sw} were obtained, and $u_K(t)$ was approximated exactly. And, as a result, accurate state approximations were obtained. Note that two superelements were sufficient to solve this problem exactly. A few runs were tried for $NSE = 3$ from various starting points, ξ^0_{ise}, although only one solution is presented here. Interestingly, but not too surprisingly, the location of the "extra" breakpoint is not unique. It has no effect on a problem that has only one discontinuity. Therefore it appears that to solve problems which may contain discontinuities, NSE needs to be sufficiently large so that all points of discontinuity can be found. This parallels the well-known fact that the goodness of an approximation obtained with finite elements increases up to a point where little more accuracy is achievable.

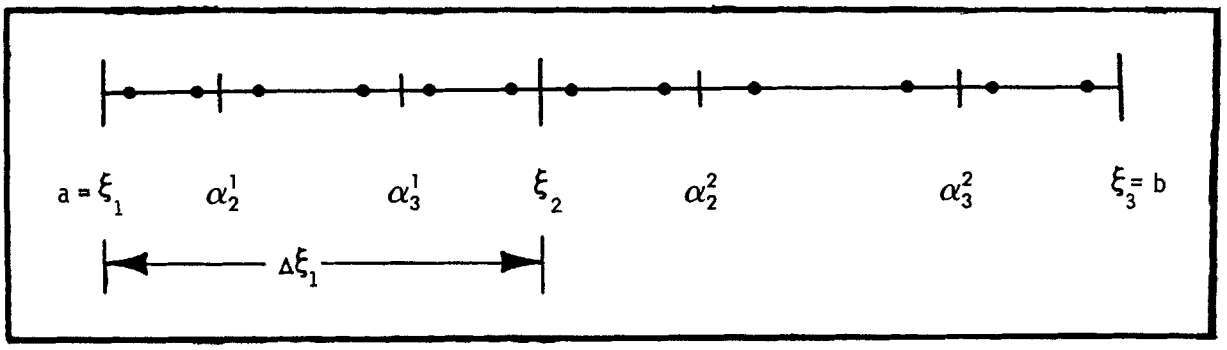

Figure 6. Superelement structure.

Table 2. Results for Problem P1 Using Superelements

NSE	Starting Point ξ^0_{ise} $ise = 2, \ldots, NSE + 1$	Optimum ξ^*_{ise} $ise = 2, \ldots, NSE + 1$	State Errors $\|Z_1^{err}(t)\|$	State Errors $\|Z_2^{err}(t)\|$
2	5, 10	20, 30	$<10^{-24}$	$<10^{-26}$
3	30, 40, 50	12.66, 20, 30	$<10^{-17}$	$<10^{-18}$

Optimization of a Reactor Problem with Adaptive Knot Placement

In this section we present the optimization of a reactor problem using NLP4. This problem contains no control profiles and no discontinuities and hence requires no superelements. However, a number of finite elements are necessary to approximate a temperature profile that contains a hot spot. We also show, through this example, that imposition of bounds on continuous profiles can be done easily by using bounds on the coefficients.

Example problem P2

The reactor optimization problem is illustrated in Figure 7. Here a 3:1 ratio of reactants B/A is first preheated by reactor effluent. This stream is then fed to a packed-bed reactor where the reaction $A + 3B \rightarrow C + 3D$ occurs. The reactor is jacketed to allow the heat of reaction to be used to raise steam for the rest of the process.

The differential-algebraic optimization problem for this reactor design is:

$$\min_{T_P, T_R, L, T_S, q(t), \bar{T}(t)} \quad \Phi = L - \int_0^L [\bar{T}(t) - T_S/T_R]\, dt$$

$$\text{s.t.} \quad \frac{dq(t)}{dt} = 0.3[1 - q(t)] \exp[20 - 20/\bar{T}(t)], \quad q(0) = 0$$

$$\frac{d\bar{T}(t)}{dt} = -1.5[\bar{T}(t) - T_S/T_R] + \frac{2}{3}\frac{dq(t)}{dt}, \quad \bar{T}(0) = 1; t \in [0, L]$$

$$C_{\text{preheater}} = \Delta H_{\text{feed}}(T_R, 110°\text{C}) - \Delta H_{\text{product}}[T_P, T(L)] = 0$$

$$T_P \geq 120°\text{C}, \quad T(L) \geq T_R + 10°\text{C}$$

$$\text{with} \quad \bar{T}(t) = T(t)/T_R \qquad \text{(P2)}$$

where T_P = specified product temperature, T_R = reactor inlet and reference temperature, L = reactor length, T_S = steam sink temperature, $q(t)$ = reactor conversion profile for species A, $\bar{T}(t)$ = normalized reactor temperature profile. This optimization involves maximizing the production of steam (utility profit) from a reactor while minimizing its length (capital cost). For the examples presented in this paper an equal normalized unit price was used to weight each term in the objective function. Similar results, however, were obtained with other weightings. The optimization of problem P2 is done with respect to four decision variables, T_P, T_R, L, and T_S, as well as the two continuous profiles, $q(t)$ and $\bar{T}(t)$. The reactor model consists of two ODE state equations, one for conversion of A and one for normalized temperature. More details about the reactor model can be found in Finlayson (1971) and Hlavacek (1970). Under the conditions given in Tables 3 and 4, for cases Ia and Ib the reactor model exhibits a hot spot while for case II no hot spot exists. Normally, when minimizing capital cost a hot spot is undesirable since beyond the hot spot the reaction rate is essentially zero. (The existence of a hot spot is really a manifestation of an overly long reactor.) However, since steam is raised, better designs can exist at longer reactor lengths. An additional process constraint appears in the form of a feed preheater heat balance. This constraint merely equates the enthalpy changes between the reactor feed and reactor effluent over the indicated temperature ranges. Two other temperature constraints are written to prevent temperature crossover in the preheater.

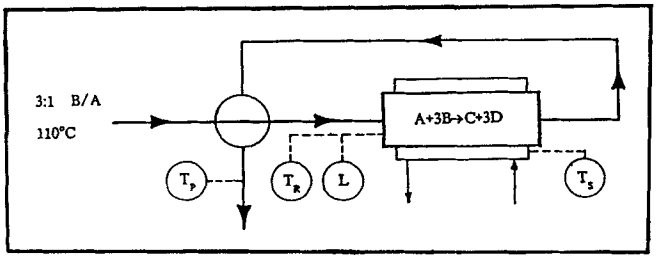

Figure 7. Flowsheet, problem P2.

These two cases (with and without a hot spot) will be discussed in more detail in the next two sections. Initialization of the continuous profiles, however, is an important consideration for these problems since it can affect significantly the performance of the optimizer and the knot placement strategy. We treat this topic in the following section.

Initialization procedure for reactor optimization

In order to illustrate the knot placement strategy and NLP4, we consider the following cases as optimization problems:

Case Ia. Reactor optimization problem P2 with a hot spot appearing in the temperature profile

Case Ib. Same as case Ia, with the additional state profile constraint $\bar{T}(t) \leq 1.45$ imposed

Case II. Reactor optimization problem P2 without a hot spot. Although this case used the same reactor model, the hot spot was avoided by using a different starting point and imposing different constraints (e.g., a shorter length).

The decision variable and profile bounds used for these cases can be found in Table 3.

Recall from the section on the development of knot placement equations that in order to obtain good knot placement the approximations $z^i_{K+1}(t)$ $\forall$ i must be reasonably close to $Z(t)$. This requires that good guesses be used for the initial approximations to $q(t)$ and $\bar{T}(t)$. For cases I and II the starting profiles were different and therefore we discuss them separately.

Table 3. Initial Bounds for Problem P2

	Case Ia		Case Ib		Case II	
	Lower	Upper	Lower	Upper	Lower	Upper
T_R, °C	400	500	400	500	400	500
L	0.5	1.25	0.5	1.25	0.5	1.0
T_s, °C	400	500	400	500	400	500
T_P, °C	100	1,000	100	1,000	100	1,000
$q(t)$	0	1.5	0	1.5	0	1.0
$T(t)$	0	3.0	0	1.45	1.0	3.0

In both case Ia and case Ib steep profiles exist and care must be taken in initializing the optimization problem. For the initial decision variable sets given in Table 4, the model was first integrated with the ODE solver LSODE (Hindmarsh, 1980). Next, an initial set of knots α_i, $i = 2, \ldots, NE$, with $\alpha_1 = 0$ and $\alpha_{NE+1} = L$, were fixed and the collocation points calculated using Eq. 7. State coefficients at the collocation points were then obtained from the integrated profiles with the coefficients at the knots obtained by solving the linear system of continuity Eq. 14. Note that since the linear system of continuity equations is solved initially, any subsequent Newton or SQP procedure always remains in the feasible subspace of these constraints (assuming that no infeasible QP subproblems exist). Finally, the residual, knot continuity, and knot placement equations were completely converged prior to starting the optimization. This was done with Newton's method starting from an evenly spaced set of knots and resulted in the knot distribution listed in Table 6. The resulting continuous approximations are superimposed upon the integrated profiles in Figure 8. For these approximations the maximum deviation from the integrated profiles is 0.02707 for conversion and 0.01745 for normalized temperature. This additional effort was required since in cases Ia and Ib existence of the hot spot makes doing the approximation more difficult.

For case II the profiles do not contain nondifferentiabilities and thus are easier to approximate. Consequently the initialization for this problem did not require the additional initial convergence of the knot placement, continuity, and collocation equations with Newton's method. Here it was sufficient to initialize the coefficients at the collocation points from an integrated profile and solve the continuity equations to obtain the coefficients at the knots.

Table 4. Initial Decision Variables and Objective Function for Problem P2

	Cases Ia, Ib	Case II
T_R, °C	462.23	434.80
L	1.0	0.67
T_s, °C	425.25	400
T_P, °C	250	400
Φ	−120.702	−45.544

Table 5. Results for Problem P2 Solved with NLP6

	Case Ia	Case Ib	Case II
T_R, °C	500	500	500
L	1.25	1.25	1.0
T_s, °C	473.8	449.9	448.4
T_P, °C	192.3	237.1	344.9
$\|\| q^{err}(t) \|\|$	0.01103	0.01129	0.02116
$\|\| T^{err}(t) \|\|$	0.007113	0.006999	0.01437
K − T error	10^{-14}	10^{-14}	10^{-14}
Φ	−171.438	−145.644	−82.7024
No. SQP iterations	28	18	24
CPU* time, s	262	162	228

*DEC-20

Results and discussion of problem P2

In this section we present the results obtained from solving P2 using formulation NLP4 and the SQP algorithm described in Biegler and Cuthrell (1985). Table 5 presents the optimal values of the decision variables, and some diagnostic information. The error norms represent the max difference over $t \in [0, L]$ between a profile constructed using Lagrange polynomials (see Eq. 9) and one obtained *a posteriori* through numerical integration. Also, the "No. SQP iterations" row refers to the number of QP subproblems solved using SQP, and "K-T error" represents the final Kuhn-Tucker error. In Table 6 a comparison is made between the initial and final knot distributions.

Cases Ia and Ib represent formulations for which the solution has a hot spot. From the solutions the length and inlet reactor temperature clearly indicate that the utility term in the objective function dominates. Since both variables are at their upper bounds the reactor is producing as much steam as possible. This condition is preferred over one with a shorter (and less expensive) reactor. Note also from Figures 8 to 10 how the hot spot shifts in size and location from the initial profile as a result of the optimization. Again, in these figures the approximated and integrated profiles are superimposed. Good approximations are obtained for the continuous profiles in both cases Ia and Ib with the max norms for conversion and temperature errors being around 0.01 or less. Table 6 clearly demonstrates the effectiveness of the knot placement equations. For cases Ia and Ib the final knots are grouped in the region of the hot spot where both state profiles are very steep.

In case Ib imposition of the state profile bound $\overline{T}(t) \leq 1.45$ lowered the peak of the temperature profile and moved it to a later position in the reactor. This is partially due to a lower sink

Table 6. Initial and Final Knot Distributions for Problem P2

	Case Ia	Case Ib	Case II
α_i initial	0, 0.365, 0.534, 0.591	same as Ia	0, 0.201, 0.268, 0.335
$i = 1, \ldots, NE + 1$	0.636, 0.692, 1.0		0.402, 0.469, 0.67
α_i final	0, 0.287, 0.416, 0.456	0, 0.486, 0.728, 0.828	0, 0.369, 0.624, 0.771
$i = 1, \ldots, NE + 1$	0.489, 0.528, 1.25	0.908, 1.00, 1.25	0.852, 0.924, 1.0

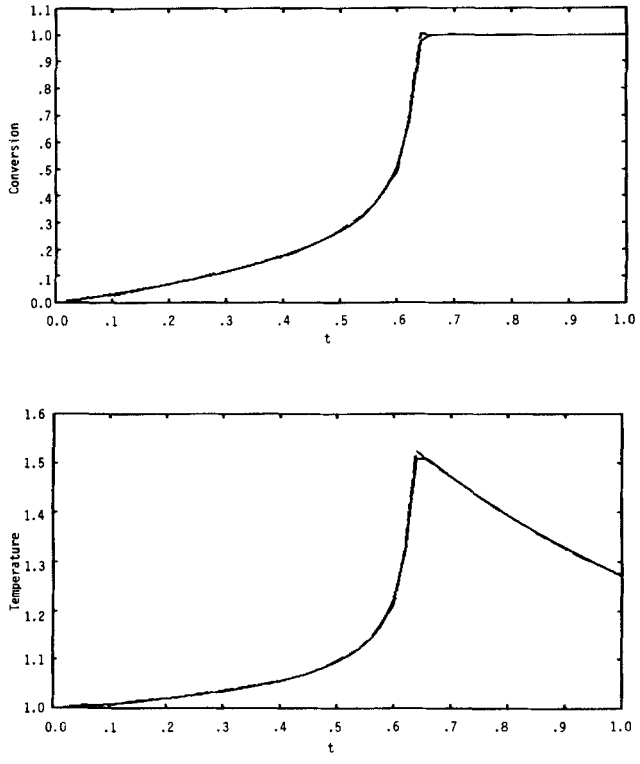

Figure 8. Starting conversion and temperature profiles, problem P2, cases Ia and Ib.

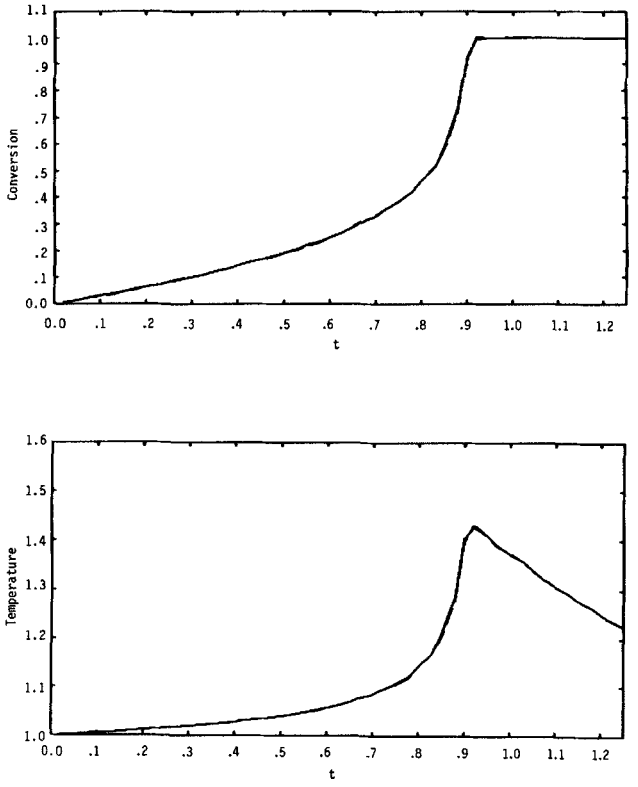

Figure 10. Optimal conversion and temperature profiles, problem P2, case Ib.

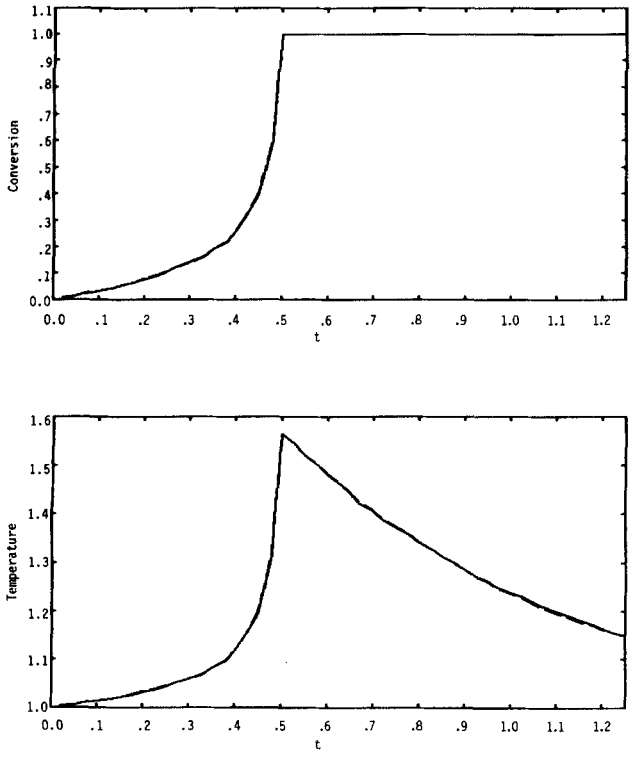

Figure 9. Optimal conversion and temperature profiles, problem P2, case Ia.

temperature (449.9°C for case Ib vs. 473.8°C for case Ia) which reduced the rate of heat removal from the reactor. More important, however, the optimal knot locations are near the region $t \in [0.7, 1.0]$ of steep profiles and thus good approximations again result.

In case II the hot spot does not exist, Figure 11. Nevertheless good approximations and good knot placement were again obtained. Here the knots are grouped in the region $t \in [0.6, 1.0]$ where both profiles are steep. Since a large part of the reactor for case I represents unnecessary capital expense it was instructive to investigate a design without a hot spot. The solution for this case, however, again has both inlet temperature and length at their upper bounds and indicates that the desire to raise steam outweighs any capital cost influence. Comparison of objective functions, $\Phi = -171.438$ vs. $\Phi = -82.7024$, therefore shows case Ia is preferred over case II.

Summary and Conclusions

This paper has presented a general and accurate method for optimizing differential-algebraic systems of equations. The differential equations are discretized by using Lagrange form polynomials and orthogonal collocation. The resulting algebraic approximations are then written as constraints in a nonlinear program. Accuracy of these approximations is guaranteed by additionally solving within the NLP the sufficient conditions for error minimization. These conditions are written as a set of knot placement constraints that allow the knot locations to be chosen adaptively. An extra level of elements, superelements, has also been developed in order to solve problems containing control

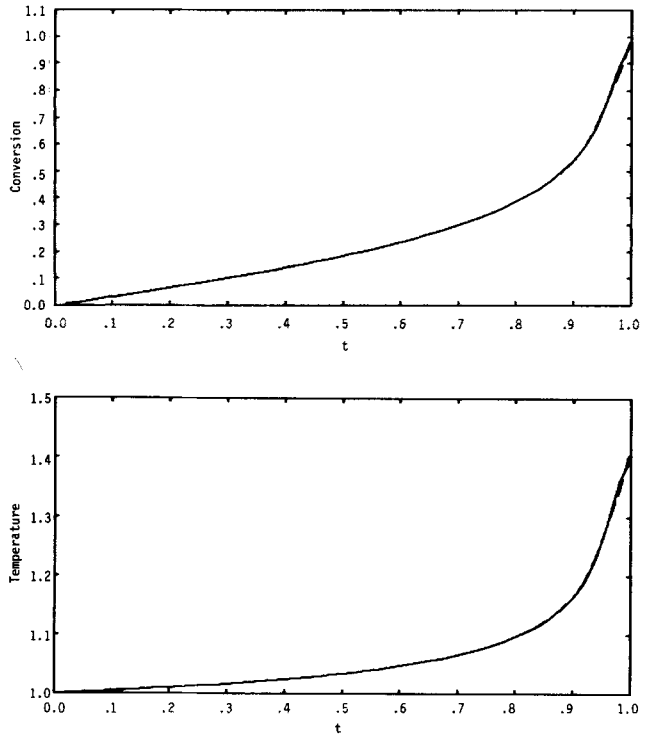

Figure 11. Optimal conversion and temperature profiles, problem P2, case II.

profile discontinuities. Location of the superelement breakpoints is left as a decision in the optimization problem.

Two examples were presented to illustrate the usefulness of the method. First, a small optimal control problem was solved using superelements and quadratic state profile approximations. Both the discontinuous (bang-bang) control profile and the state profiles were approximated extremely well. For this problem the need to use finite elements was avoided since the analytic solutions were known to be linear and quadratic polynomials. Second, a reactor optimization problem was presented to illustrate that accurate approximations can be obtained within the context of optimization by using finite-element collocation and an error minimization strategy. This problem also illustrates that bounding of state variable profiles can be done with no additional difficulty.

In future work more efficient methods need to be developed to solve the large NLP's that result. The advantage of the approach outlined in this paper over one using repeated numerical integration was demonstrated in Cuthrell and Biegler (1985). However, for problems solved with NLP4, which require a large number of both finite elements and superelements, the task of solving the NLP is not trivial. Investigation of decomposition techniques applicable to SQP methods remains one of the future directions of this research.

Acknowledgment

Partial support for this research is made possible through an educational grant to Carnegie-Mellon Chemical Engineering Dept. from the Shell Education Foundation, and from National Science Foundation Grant No. ENG-8451058.

Appendix A: Lagrange Polynomials on Finite Elements

Equation 9 can be written in general as:

$$z^i_{K+1}(t) = \sum_{j=0}^{K} \mathscr{z}_{(i-1)(K+1)+j}\phi_{(i-1)(K+1)+j}(t)$$

$$\phi_{(i-1)(K+1)+j}(t) = \prod_{k=0,j}^{K} \frac{[t - t_{(i-1)(K+1)+k}]}{[t_{(i-1)(K+1)+j} - t_{(i-1)(K+1)+k}]}$$

for $i = 1, \ldots, NE$, which becomes for $NE = 3$ and $K = 2$:

$$z^1_3(t) = \sum_{j=0}^{2} \mathscr{z}_j\phi_j(t) \quad \phi_j(t) = \prod_{k=0,j}^{2} \frac{(t - t_k)}{(t_j - t_k)}$$

$$z^2_3(t) = \sum_{j=0}^{2} \mathscr{z}_{3+j}\phi_{3+j}(t) \quad \phi_{3+j}(t) = \prod_{k=0,j}^{2} \frac{(t - t_{3+k})}{(t_{3+j} - t_{3+k})}$$

$$z^3_3(t) = \sum_{j=0}^{2} \mathscr{z}_{6+j}\phi_{6+j}(t) \quad \phi_{6+j}(t) = \prod_{k=0,j}^{2} \frac{(t - t_{6+k})}{(t_{6+j} - t_{6+k})}$$

Appendix B: Effectiveness of Knot Placement Strategy

Here we present one additional result of the packed-bed reactor simulation problem, treated in the section on initialization procedure for reactor optimization to illustrate the effectiveness of the knot placement equations. That section lists the starting profiles used to optimize P2 that were constructed after solving the residual equations, Eq. 12, continuity relations, Eq. 14, and

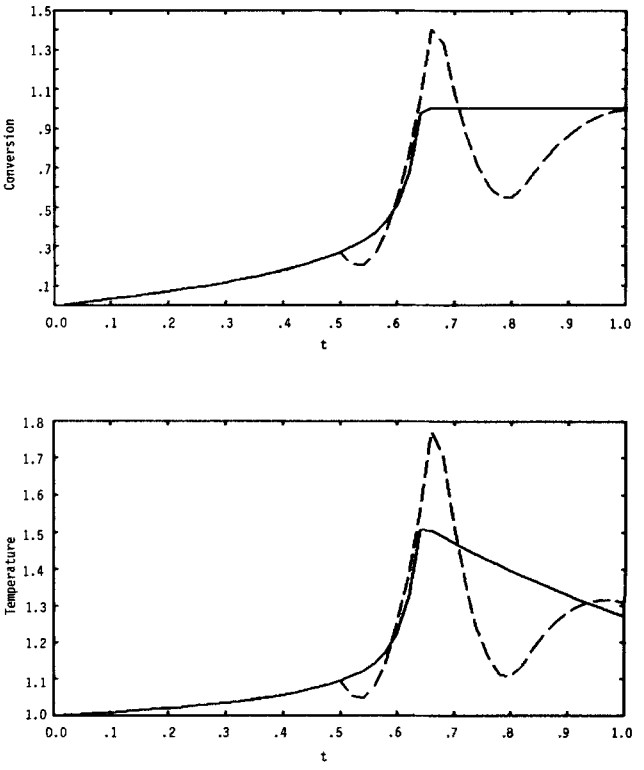

Figure 12. Conversion and temperature profiles with evenly spaced elements.

knot placement equations, Eq. 21. These profiles are depicted in Figure 8 and were obtained with six finite elements.

A similar solution of these equations was obtained with six evenly spaced finite elements; the results are depicted in Figure 12. In this case, evenly spaced knots were used instead of solving knot placement Eqs. 21. Again, in these figures the analytical solution is shown by a solid line and the approximations by a broken line. Observe that with evenly spaced elements the approximations are extremely poor. Moreover by comparing Figure 12 with Figure 8 it is clear that the knot placement equations are effective at positioning the finite-element knots so that good approximations are obtained.

Literature cited

Ascher, U., J. J. Christiansen, and R. D. Russell, "A Collocation Solver for Mixed-Order Systems of Boundary Value Problems," *Math. Comp.*, **33**(146), 659 (1979).

Bapat, S., and J. C. Heydweiller, "Adaptive Polynomial Approximations for Process Design," Paper 18b., AIChE Nat. Meet., Chicago (Nov. 1985).

Biegler, L. T., and J. E. Cuthrell, "Improved Infeasible Path Optimization for Sequential Modular Simulators. II: The Optimization Algorithm," *Comp. Chem. Eng.*, **9**(3), 257 (1985).

Bryson, A. E., and Y.-C. Ho, *Applied Optimal Control,* Hemisphere, Washington DC (1975).

Cuthrell, J. E., "On the Optimization of Differential-Algebraic Systems of Equations in Chemical Engineering," Ph.D. Diss., Carnegie-Mellon Univ., Pittsburgh (1986).

Cuthrell, J. E., and L. T. Biegler, "Simultaneous Solution and Optimization of Process Flowsheets with Differential Equation Models," *Ind. Chem. Eng. Symp. Ser. No. 92,* 13 (1985).

deBoor, C., "Good Approximation by Splines with Variable Knots. II," *Dundee Conf. Numerical Solution of Differential Equations, Lecture Notes in Math,* No. 363 (1974).

———, *A Practical Guide to Splines,* Springer-Verlag (1978).

deBoor, C., and B. Swartz, "Collocation at Gaussian Points," *SIAM J. Numer. Anal.*, **10**(4), 582 (1973).

Fiacco, A. V., "Sensitivity Analysis for Nonlinear Programming Using Penalty Methods," *Math. Prog.*, **10,** 287 (1976).

Finlayson, B. A., "Packed-bed Reactor Analysis by Orthogonal Collocation," *Chem. Eng. Sci.*, **26,** 1081 (1971).

———, *The Method of Weighted Residuals and Variational Principles,* Academic Press, New York (1972).

Gardini, L., A. Servida, M. Morbidelli, and S. Carra, "Use of Orthogonal Collocation on Finite Elements with Moving Boundaries for Fixed-Bed Catalytic Reactor Simulation," *Comp. Chem. Eng.*, **9**(1), 1 (1985).

Han, S.-P., "A Globally Convergent Method for Nonlinear Programming," *JOTA,* **22**(3), 297 (1977).

Hindmarsh, A. C., "LSODE and LSODI, Two New Initial-Value Ordinary Differential Equation Solvers," ACM-SIGNUM *Newsletter,* **15**(4), 10 (1980).

Hlavacek, V., "Aspects of Design of Packed Catalytic Reactors," *Ind. Eng. Chem.*, **62**(7), 9 (1970).

Lynn, L. L., E. S. Parkin, and R. L. Zahradnik, "Near-Optimal Control by Trajectory Approximation," *Ind. Eng. Chem. Fundam.*, **9**(1), 58 (1970).

Lynn, L. L., and R. L. Zahradnik, "The Use of Orthogonal Polynomials in the Near-Optimal Control of Distributed Systems by Trajectory Approximation," *Int. J. Control,* **12**(6), 1079 (1970).

Murtagh, B. A., and M. A. Saunders, "MINOS/Augmented Supplementary User's Manual," Rept. OR/78/6, Univ. New South Wales, Australia (1978).

Neuman, C. P., and A. Sen, "A Suboptimal Control Algorithm for Constrained Problems Using Cubic Splines," *Automatica,* **9,** 601 (1973a).

———, "Weighted Residual Methods and the Suboptimal Control of Distributed Parameter Systems," *Int. J. Control,* **18**(6), 1291 (1973b).

Oh, S. H., and R. Luus, "Use of Orthogonal Collocation Method in Optimal Control Problems," *Int. J. Control,* **26**(5), 657 (1977).

Pereyra, V., and E. G. Sewell, "Mesh Selection for Discrete Solution of Boundary Problems in Ordinary Differential Equations," *Numer. Math.*, **23,** 261 (1975).

Powell, M. J. D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Dundee Conf. Numerical Anal.*, Dundee, Scotland (1977).

Russell, R. D., and J. Christiansen, "Adaptive Mesh Selection Strategies for Solving Boundary-Value Problems," *SIAM J. Numerical Anal.*, **15**(1), 59 (1978).

Sargent, R. W. H., and G. R. Sullivan, "The Development of an Efficient Optimal Control Package," *Proc. 8th IFIP Conf. Optimization Tech., Pt. 2* (1977).

Tsang, T. H., D. M. Himmelblau, and T. F. Edgar, "Optimal Control via Collocation and Nonlinear Programming," *Int. J. Control,* **21**(5), 763 (1975).

Villadsen, J., and M. L. Michelsen, *Solution of Differential Equation Models by Polynomial Approximation,* Prentice-Hall, Englewood Cliffs, NJ (1978).

Villadsen, J., and W. E. Stewart, "Solution of Boundary Value Problems by Orthogonal Collocation," *Chem. Eng. Sci.*, **22,** 1483 (1967).

Wong, K. T., and R. Luus, "Time Suboptimal Feedback Control of Systems Described by Linear Parabolic Partial Differential Equations," *Opt. Control Appl. Meth.*, **3,** 177 (1982).

Manuscript received June 30, 1986, and revision received Dec. 12, 1986.