Global Optimization for the Parameter Estimation of Differential-Algebraic Systems

William R. Esposito and Christodoulos A. Floudas*
Department of Chemical Engineering
Princeton University
Princeton, N.J. 08544-5263
Tel: (609) 258-4595
Fax: (609) 258-0211
email: floudas@titan.princeton.edu

Abstract

The estimation of parameters in semi-empirical models is essential in numerous areas of engineering and applied science. In many cases these models are represented by a set of nonlinear differential-algebraic equations. This introduces difficulties from both a numerical and an optimization perspective. One such difficulty, which has not been adequately addressed, is the existence of multiple local minima.

In this paper, two novel global optimization methods will be presented which offer a theoretical guarantee of convergence to the global minimum for a wide range of problems. The first is based on converting the dynamic system of equations into a set of algebraic constraints through the use of collocation methods. The reformulated problem has interesting mathematical properties which allow for the development of a deterministic branch and bound global optimization approach. The second method is based on the use of integration to solve the dynamic system of equations. Both methods will be applied to the problem of estimating parameters in differential-algebraic models through the *error-in-variables* approach. The mathematical properties of the formulation which lead to specialization of the algorithms will be discussed. Then, the computational aspects of both approaches will be presented and compared through their application to several problems involving reaction kinetics.

*Author to whom all correspondence should be addressed

1 Introduction

The estimation of parameters in kinetic expressions from time series data is essential for the design, optimization, and control of many chemical systems. The models that describe the kinetics take the form of a set of differential-algebraic equations. The statistics and formulation of this parameter estimation problem are well studied[1;2], but the important issue of multiplicity of local solutions has not been well addressed. Stewart et al.[3] provide a review of various approaches for multi-response models, including dynamical systems, and available software implementations. Most of the difficulties in the algorithmic framework result from the dynamic form of the model used. Several techniques, ranging from numerical integration, to polynomial approximation in time have been used to handle the differential-algebraic nature of the models.

In general, there are two different direct approaches to addressing this problem. In each approach the objective is to minimize a weighted distance measure between the observations and predictions taken from the model. The first, more traditional approach, involves the use of integration routines to determine the values of the states for a given set of model parameter values. This in turn allows for the evaluation of the objective function and its gradients. This approach is referred to as a sequential algorithm, since the optimization steps and the solution of the differential system are performed in a sequential manner. Bellman et al.[4] presented a method of this type using a quasi-linearization procedure. Hwang and Seinfeld[5] proposed an improved algorithm which uses the sensitivities of the predicted states with respect to the parameters. They also noted that this formulation has the possibility of multiple minima and derived a method of improving the chance of finding the global minimum by adjusting the objective function weighting. Bilardello et al.[6] presented a similar approach using a Gauss-Newton type step with the inclusion of explicitly derived and efficiently evaluated sensitivity equations. Kim et al.[7] presented an approach which uses a generally available integration method coupled with an available reduced-gradient NLP solver for both off-line and on-line estimation problems.

Kalogerakis and Luus[8] recognized the existence of local minima and the poor convergence regions of the previous approaches[9]. They presented an algorithm which uses a direct search method[10] to determine good initial guesses for either a quasi-linearization or a Gauss-Newton type minimization. Park and Froment[11] used the same concept, but with a genetic algorithm as the first step, followed by a Levenberg-Marquardt optimization. Luus[12] used a modified direct search method to solve a problem with a large number of local minima. Maria[13] developed an approach based on a adaptive random search for the solution of various kinetic problems. Finally, Wolf and Moros[14] presented an approach based on a genetic algorithm for the determination of rate constants in heterogeneous reaction systems.

A second type of direct approach involves the conversion of the dynamic system into a set of algebraic equations which are included directly in the formulation. Many researchers refer to this as a simultaneous approach. Villadsen and Michelsen[15] offered a wide range of polynomial approximations used to solve differential equations. Van Den Brosch and Hellinckx[16] used collocation techniques combined with a linearization based approach to solve first and second order kinetic systems. Baden and Villadsen[17] studied comparisons between different collocation methods. Tjoa and Biegler[18] presented an approach using orthogonal collocation on finite elements along with a specially developed sequential quadratic programming (SQP) algorithm. Finally, Liebman et al.[19] used the same discretization method, but applied a general SQP approach to solve both parameter estimation and data reconciliation problems.

A common limitation of all the aforementioned approaches is that they can, at best, achieve convergence to a local minimum. Some approaches attempted to increase the likelihood of finding the global minimum, but they lack a theoretical guarantee. In this paper two different approaches, one based on integrating the dynamic system, and one using orthogonal collocation on finite elements, will be presented for addressing the parameter estimation problem from the global optimization point of view. Each of these methods is based on the deterministic branch and bound global optimization algorithm αBB[20;21;22;23;24]. In each case, convergence to the global minimum is theoretically guaranteed for a wide range of problems. The formulations of the approaches will be presented, and their properties which allow for the development of the customized global optimization algorithm will be discussed. The algorithmic aspects will then be presented and each of the two approaches will be used to solve several kinetic estimation problems. The theoretical and the computational aspects of the methods, as well as comparisons between the two approaches will be presented.

2 Problem Definition

In this paper, the *error-in-variables* formulation will be used to estimate the parameters in a differential-algebraic model of the form:

$$\begin{aligned} \dot{\mathbf{z}}_j &= \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t) \quad j \in J \\ \mathbf{0} &= \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t) \\ \mathbf{z}_j(t_0) &= \mathbf{z}_0 \;\; j \in J \\ t &\in [t_0,t_f] \end{aligned} \tag{1}$$

where Z is the set of $\mathbf{z}$ state variables, J is the set of states whose derivatives appear explicitly in the model, P is the set of parameters $\boldsymbol{\theta}$ which are to be estimated and $\mathbf{z}_0$ are the constant initial conditions of size J. In the *error-in-variables* approach, the objective is to minimize the weighted squared error between the observed values and those predicted by the model. All the measured variables are included in the objective function, as opposed to just a subset employed in the least squares approach. A derivation can be found in Esposito and Floudas[25] and Bard[1]. Using the definition of the model presented in (1) and assuming that the variance of the error associated with each measured variable is equal, the resulting optimization problem can be written as:

$$
\begin{aligned}
&\min_{\boldsymbol{\theta},\,\hat{\mathbf{z}}_\mu} \quad \sum_{m\in M}\sum_{\mu=1}^{r}(\hat{z}_{\mu,m} \;-\; \bar{z}_{\mu,m})^2 \qquad (2)\\
&\text{s.t.}\\
&\qquad \dot{\mathbf{z}}_j \;=\; \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t) \quad j\in J\\
&\qquad \mathbf{0} \;=\; \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t)\\
&\qquad \mathbf{z}_j(t_0) \;=\; \mathbf{z}_0 \quad j\in J\\
&\qquad t \;\in\; [t_0,t_f]\\
&\qquad -\hat{z}_{\mu,m} \;+\; z_m(t_\mu) \;=\; 0 \quad m\in M; \quad \mu=1,\dots,r\\
&\qquad \boldsymbol{f}(\hat{\mathbf{z}}_\mu,\boldsymbol{\theta}) \;\le\; \mathbf{0}
\end{aligned}
$$

where $\hat{\mathbf{z}}_\mu$ is the set M of fitted data variables, $\bar{\mathbf{z}}_\mu$ are the observed values at the μ^{th} data point (includes a total of r points), and t_μ is the time associated with the μ^{th} data point. The functions $\boldsymbol{f}(\hat{\mathbf{z}}_\mu,\boldsymbol{\theta})$, are a set of algebraic constraints which may only involve the time invariant variables (the fitted data variables, $\hat{\mathbf{z}}_\mu$, and the parameters, $\boldsymbol{\theta}$). It should be noted that $M \subseteq Z$ and also $J \subseteq Z$. The following conditions are imposed:

1. The differential-algebraic system can be at most index one, and therefore the Jacobian defined by

$$\frac{\partial \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t)}{\partial z_i} \qquad i \notin J$$

must be non-singular in the region $\mathbf{z}\in[\mathbf{z}^L,\mathbf{z}^U]$.

2. The second order partial derivatives:

$$\frac{\partial^2 \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t)}{\partial \mathbf{x}^2}\;,\quad \frac{\partial^2 \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t)}{\partial \mathbf{x}^2}$$

where $\mathbf{x} \equiv [\mathbf{z},\ \boldsymbol{\theta}]$ must exist and be continuous in the region defined by $\mathbf{z}\in[\mathbf{z}^L,\mathbf{z}^U]$ and $\boldsymbol{\theta}\in[\boldsymbol{\theta}^L,\boldsymbol{\theta}^U]$.

3. The functions $\boldsymbol{f}(\hat{\mathbf{z}}_\mu,\boldsymbol{\theta})$ must be twice-continuously differentiable with respect to the variables $\hat{\mathbf{z}}_\mu$ and $\boldsymbol{\theta}$ in the region defined by $\hat{\mathbf{z}}_\mu\in[\hat{\mathbf{z}}_\mu^L,\hat{\mathbf{z}}_\mu^U]$ and $\boldsymbol{\theta}\in[\boldsymbol{\theta}^L,\boldsymbol{\theta}^U]$.

3 Collocation Based Approach

Many different collocation based discretizations exist for the solution of differential-algebraic systems[26]. In this section, the method of orthogonal collocation on finite elements will be used to parameterize the state variables[27;28]. This transforms the original problem into a fully algebraic NLP.

3.1 Formulation

The differential-algebraic part of (2) is defined as:

$$\begin{aligned} \dot{\mathbf{z}}_j &= \boldsymbol{g}(\mathbf{z}, \boldsymbol{\theta}, t) \quad j \in J \\ \mathbf{0} &= \boldsymbol{h}(\mathbf{z}, \boldsymbol{\theta}, t) \ . \end{aligned} \tag{3}$$

The state profiles are approximated by piecewise Lagrange polynomials:

$$\mathbf{z}_{K+1}(t) = \sum_{c=0}^{K} \boldsymbol{\xi}_c \, \phi_c(t) \quad \text{with} \quad \phi_c(t) = \prod_{q=0, q \neq c}^{K} \frac{t - t_q}{t_c - t_q} \ . \tag{4}$$

These polynomials have the feature that at the time point t_c, the coefficient of the polynomial, ξ_c, is the value of the state profile at that point. Therefore, the coefficients have a physical meaning which makes it easier to generate bounds for these variables. Substituting (4) into (3) yields the residual equations:

$$\mathbf{R}(t) \ = \ \sum_{c=0}^{K} \boldsymbol{\xi}_c \, \dot{\phi}_c(t) \ - \ \boldsymbol{g}(\mathbf{z}_{K+1}(t), \boldsymbol{\theta}, t) \ . \tag{5}$$

The discretization of (5) is accomplished using the collocation method[26]:

$$\int_0^{t_f} \mathbf{R}(t) \, \delta(t - t_k) \, dt \ = \ 0 \qquad k = 1, \ldots, K \ . \tag{6}$$

The time variable, t, is rescaled as $\tau \in [0, 1]$. Using this rescaling, and the property of Lagrange polynomials stated above, the integral reduces to:

$$\sum_{c=0}^{K} \boldsymbol{\xi}_c \, \dot{\phi}_c(\tau_k) \ - \ t_f \, \boldsymbol{g}(\boldsymbol{\xi}_k, \boldsymbol{\theta}, t_k) \ = \ 0 \qquad k = 1, \ldots, K \tag{7}$$

where $t_k = t_f \tau_k$. The values of scaled time at which this equality is imposed, τ_k, are referred to as collocation points. The shifted roots of an orthogonal Legendre polynomial of degree K are used.

In this approach, the polynomial approximation is defined for each finite element (see Figure 1), $e = 1, \ldots, NE$, and for each state variable in the set J. The residual equations are then written as:

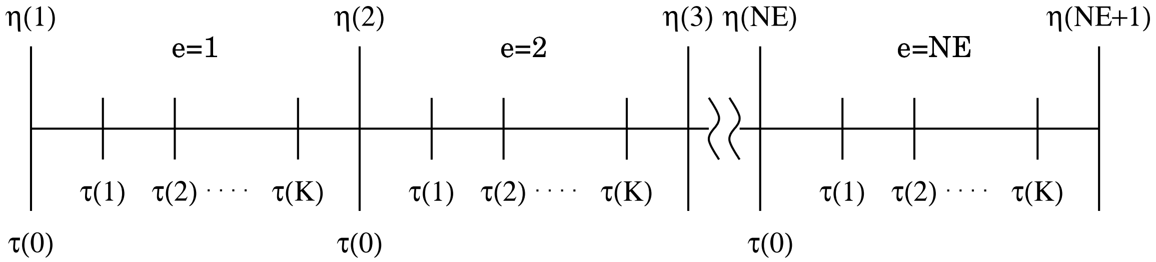


Figure 1: Finite Element Collocation.

$$\sum_{c=0}^{K} \xi_{e,c,j}\ \dot{\phi}_{e,c,j}(\tau_k)\ -\ \Delta\eta_e\, g_j(\xi_{e,k,j}, \boldsymbol{\theta}, t_{e,k})\ =\ 0 \qquad (8)$$
$$e = 1, \ldots NE; \qquad k = 1, \ldots, K; \qquad j \in J$$

where $\Delta\eta_e = \eta_{e+1} - \eta_e$ with η_e being the starting time of element e, and $t_{e,k} = \eta_e + \Delta\eta_e\, \tau_k$. Since the state profiles should be continuous between elements, continuity constraints of the form:

$$\xi_{e,0,j}\ -\ \sum_{c=0}^{K} \xi_{e-1,c,j}\, \phi_c(\tau = 1)\ =\ 0 \qquad e = 2, \ldots, NE; \qquad j \in J \qquad (9)$$

are imposed. In addition, the initial conditions are imposed at the start of the first element:

$$\xi_{1,0,j}\ -\ z_{0,j}\ =\ 0 \qquad j \in J\ . \qquad (10)$$

The algebraic constraints of the differential-algebraic system are simply imposed at the collocation points:

$$\boldsymbol{h}(\boldsymbol{\xi}_{e,k}, \boldsymbol{\theta}, t_{e,k})\ =\ \mathbf{0} \qquad e = 1, \ldots, NE; \quad k = 1, \ldots, K\ . \qquad (11)$$

To introduce the point constraints, which determine the fitted values of the data variables, $\hat{z}_\mu$, it is necessary to determine the values of the states at the times t_μ. This is accomplished by:

$$z_m(t_\mu) = \sum_{c=0}^{K} \xi_{e_\mu,c,m}\phi_c(\tau_\mu) \qquad m \in M\ . \qquad (12)$$

The scaled time, τ_μ, is calculated by

$$\tau_\mu \ = \ \frac{t_\mu - \eta_{e_\mu}}{\Delta \eta_{e_\mu}} \tag{13}$$

where the element index, e_μ, is the element in which the constraint is active, defined as $e_\mu \equiv \{e : \eta_e \leq t_\mu < \eta_{e+1}\}$. The constraints are then rewritten as:

$$-\hat{z}_{\mu,m} \ + \ \sum_{c=0}^{K} \xi_{e_\mu,c,m} \phi_c(\tau_\mu) \ = \ 0 \quad m \in M; \ \ \mu = 1, \ldots, r \ \ . \tag{14}$$

Then, the complete formulation is:

$$\begin{aligned}
&\min_{\hat{\mathbf{z}}_\mu, \boldsymbol{\xi}, \boldsymbol{\theta}} \ \sum_{m \in M} \sum_{\mu=1}^{r} (\hat{z}_{\mu,m} \ - \ \bar{z}_{\mu,m})^2 \\
&\text{s.t.} \\
&\quad \sum_{c=0}^{K} \xi_{e,c,j} \, \dot{\phi}_{e,c,j}(\tau_k) - \Delta\eta_e \, g_j(\xi_{e,k,j}, \boldsymbol{\theta}, t_{e,k}) = 0 \quad e = 1, \ldots, NE; \ k = 1, \ldots, K; \ j \in J \\
&\quad \boldsymbol{h}(\boldsymbol{\xi}_{e,k}, \boldsymbol{\theta}, t_{e,k}) \ = \ \mathbf{0} \quad e = 1, \ldots, NE; \ k = 1, \ldots, K \\
&\quad \xi_{e,0,j} \ - \ \sum_{c=0}^{K} \xi_{e-1,c,j} \, \phi_c(\tau = 1) \ = \ 0 \quad e = 2, \ldots, NE; \ j \in J \\
&\quad \xi_{1,0,j} \ - \ z_{0,j} \ = \ 0 \quad j \in J \\
&\quad -\hat{z}_{\mu,m} \ + \ \sum_{c=0}^{K} \xi_{e_\mu,c,m} \phi_c(\tau_\mu) \ = \ 0 \quad m \in M; \ \ \mu = 1, \ldots, r \\
&\quad \boldsymbol{f}(\hat{\mathbf{z}}_\mu, \boldsymbol{\theta}) \ \leq \ \mathbf{0} \ \ .
\end{aligned} \tag{15}$$

The formulation (15) exhibits the following properties:

1. The size of the optimization domain is greatly increased by the inclusion of the collocation variables, $\boldsymbol{\xi}$. For instance, a problem with 3 states, which is approximated using third order polynomials on 5 elements, adds 45 collocation variables.

2. The choice of the number of elements, the placement of those elements, and the degree of the polynomial used have a significant impact on the error imposed by the approximation.

3. Even though the number of variables has been increased, the number of degrees of freedom has not changed. All the additional collocation variables appear in equality constraints.

4. The forms of the nonlinearities in the problem have not changed through the collocation, and additional nonlinearities have not been introduced. The polynomials used are linear in the variables $\boldsymbol{\xi}$, and hence the additional constraints introduced for continuity, and initial conditions are linear. Furthermore, the first term in the residual equations and the variable terms in the point constraints are also linear. Although this is the case, the number of nonlinear variables is greatly increased. Using the illustration of property 1, of the 45 additional collocation variables, 30 could appear in nonlinear terms depending on the form of the functions $\boldsymbol{g}$ and $\boldsymbol{h}$.

5. Since the problem is now algebraic, and twice differentiable in nature, reliable and deterministic methods for the generation of the global minimum exist (e.g., the αBB algorithm[23;24]).

3.2 Basic Ideas of the αBB

The αBB global optimization method[20;21;22;23;24] guarantees convergence to the global minimum for general twice continuously differentiable constrained and unconstrained NLPs. This is accomplished through the generation of a non-decreasing sequence of lower and a non-increasing sequence of updated upper bounds on the global solution. Finite ϵ-convergence to the global minimum is achieved through the successive subdivision of the region at each level in the branch and bound tree. The sequence of upper bounds on the global solution is obtained by solving, to local optimality, the full nonconvex problem from different starting points. The lower bounds are generated by solving a convex relaxation which underestimates the original problem. The key to the success of the αBB is its ability to generate an underestimating formulation. The formulation needs to be relaxed enough to ensure convexity, but not too much so as to avoid overly underestimating the original problem. The recent book by Floudas[29] presents the theoretical advances and a variety of computational methods and applications in the area of global optimization.

3.3 Underestimating Formulation

Since the formulation given by (15) is algebraic in nature, the development of the underestimator is straight forward. First, the formulation needs to be reduced to a standard form in which all the nonlinear terms appear in inequality constraints. This is accomplished by splitting each equality constraint into a positively and a negatively signed inequality. Then, all the nonconvex terms in (15) are underestimated separately depending on their form. For instance, the tightest convex relaxation of a univariate concave term, $f(x)$, from x^U to x^L is given by the linearization:

$$\mathcal{L}(x) \;=\; f(x^L) \;+\; \frac{f(x^U) \;-\; f(x^L)}{x^U \;-\; x^L}\,(x \;-\; x^L) \;\; . \tag{16}$$

Property 1 *The maximum separation distance between a univariate concave term and its underestimator occurs at the point $\hat{x}$ in the region defined by $[x^U, x^L]$ where the following condition is met:*

$$\frac{df}{dx} - \frac{f(x^U) \; - \; f(x^L)}{x^U \; - \; x^L} \; = \; 0, \tag{17}$$

and takes the value of:

$$f(\hat{x}) \; - \; f(x^L) \; - \; \frac{f(x^U) \; - \; f(x^L)}{x^U \; - \; x^L} \, (\hat{x} \; - \; x^L) \quad . \tag{18}$$

Proof. The maximum separation distance between the original function and the underestimator is defined by the following problem:

$$\begin{aligned} \max_x \quad & f(x) \; - \; \mathcal{L}(x) \\ \text{s.t.} \quad & x^L \; \leq \; x \; \leq \; x^U \quad . \end{aligned} \tag{19}$$

Since $f(x)$ is concave and $\mathcal{L}(x)$ is linear, the above problem is concave over the entire region, $[x^L, x^U]$. Therefore there exists one unique maximum to the problem. Substituting the underestimator into (19) results in:

$$\begin{aligned} \max_x \quad & f(x) \; - \; f(x^L) \; - \; \frac{f(x^U) \; - \; f(x^L)}{x^U \; - \; x^L} \, (x \; - \; x^L) \\ \text{s.t.} \quad & x^L \; \leq \; x \; \leq \; x^U \quad . \end{aligned} \tag{20}$$

The unique maximum of (20) occurs when the derivative with respect to x is set equal to zero:

$$\frac{df(x)}{dx} \; - \; \frac{f(x^U) \; - \; f(x^L)}{x^U \; - \; x^L} \; = \; 0 \quad . \tag{21}$$

Remark 1 The value of the maximum separation and the point at which it occurs is specific to the term under study.

Remark 2 The maximum separation distance occurs at the point where the tangent to the original function is parallel to the underestimator.

Illustration. Consider the term $-x^2$. The condition given by (17) results in:

$$-2\,x \;+\; \frac{(x^U)^2 \;-\; (x^L)^2}{x^U \;-\; x^L} \;=\; 0 \;\; . \tag{22}$$

Solving for x, the maximum separation occurs at the point,

$$\hat{x} \;=\; \frac{1}{2}(x^U + x^L)$$

with a value of,

$$\frac{1}{4}(x^U \;-\; x^L)^2 \;\; .$$

Two types of terms which appear often in kinetic estimation problems are the bilinear term, xy, and the trilinear term, xyz. The bilinear terms will be relaxed using a method developed by Al-Khayyal and Falk[30] and McCormick[31]. The term xy is replaced by an auxiliary variable w, and the following four linear inequality constraints are added:

$$\begin{array}{rcl} x^L y + y^L x - x^L y^L - w & \leq & 0 \\ x^U y + y^U x - x^U y^U - w & \leq & 0 \\ -x^U y - y^L x + x^U y^L + w & \leq & 0 \\ -x^L y - y^U x + x^L y^U + w & \leq & 0 \;\; . \end{array} \tag{23}$$

The first two constraints given in (23) produce an underestimator of the term, while the second two generate the overestimator. The convex relaxation for a trilinear term is derived from the cuts for a bilinear term given by (23). Each term xyz is also replaced by an auxiliary variable w, and the following 16 linear inequality constraints are added:

$$
\begin{array}{rcl}
xy^Lz^L + x^Lyz^L + x^Ly^Lz - x^Ly^Lz^L - x^Ly^Lz^L - w & \leq & 0 \\
xy^Uz^U + x^Uyz^U + x^Uy^Uz - x^Uy^Uz^U - x^Uy^Uz^U - w & \leq & 0 \\
xy^Lz^L + x^Lyz^U + x^Ly^Uz - x^Ly^Lz^L - x^Ly^Uz^U - w & \leq & 0 \\
xy^Uz^U + x^Uyz^L + x^Uy^Lz - x^Uy^Uz^U - x^Uy^Lz^L - w & \leq & 0 \\
xy^Uz^L + x^Uyz^L + x^Ly^Lz - x^Ly^Lz^L - x^Uy^Uz^L - w & \leq & 0 \\
xy^Lz^U + x^Lyz^U + x^Uy^Uz - x^Ly^Lz^U - x^Uy^Uz^U - w & \leq & 0 \\
xy^Lz^U + x^Lyz^L + x^Uy^Lz - x^Ly^Lz^L - x^Uy^Lz^U - w & \leq & 0 \\
xy^Uz^L + x^Uyz^U + x^Ly^Uz - x^Ly^Uz^L - x^Uy^Uz^U - w & \leq & 0 \\
-xy^Uz^U - x^Lyz^U - x^Ly^Lz + x^Ly^Lz^U + x^Ly^Uz^U + w & \leq & 0 \\
-xy^Uz^U - x^Lyz^L - x^Ly^Uz + x^Ly^Uz^L + x^Ly^Uz^U + w & \leq & 0 \\
-xy^Lz^L - x^Uyz^U - x^Uy^Lz + x^Uy^Lz^U + x^Uy^Lz^L + w & \leq & 0 \\
-xy^Lz^L - x^Uyz^L - x^Uy^Uz + x^Uy^Uz^L + x^Uy^Lz^L + w & \leq & 0 \\
-xy^Lz^U - x^Uyz^U - x^Ly^Lz + x^Ly^Lz^U + x^Uy^Lz^U + w & \leq & 0 \\
-xy^Lz^L - x^Uyz^U - x^Uy^Lz + x^Uy^Lz^L + x^Uy^Lz^U + w & \leq & 0 \\
-xy^Uz^U - x^Lyz^L - x^Ly^Uz + x^Ly^Uz^U + x^Ly^Uz^L + w & \leq & 0 \\
-xy^Uz^L - x^Lyz^L - x^Uy^Uz + x^Uy^Uz^L + x^Ly^Uz^L + w & \leq & 0 \ .
\end{array}
\tag{24}
$$

Note that the first eight inequalities in (24) correspond to the underestimator, while the last eight represent the overestimator. Other special types of underestimators (e.g., for fractional, signomial, products of univariate terms, etc.) can be found in Esposito and Floudas[25], Adjiman et al.[23], Maranas and Floudas[32], Quesada and Grossmann[33], and Zamora and Grossmann[34]. For general nonconvex terms a novel underestimation scheme was developed by Maranas and Floudas[35]. For a given general nonconvex term in several variables, $NC(\mathbf{x})$, the underestimator (with a uniform diagonal shift approach[23]) is written as:

$$
\mathcal{L}(\mathbf{x}) \; = \; NC(\mathbf{x}) \; + \; \alpha \sum_{i \, \in \, \mathcal{X}} (x_i^U \; - \; x_i) \, (x_i^L \; - x_i) \tag{25}
$$

where $\mathcal{X}$ is the set of $\mathbf{x}$ variables participating in the term $NC(\mathbf{x})$. The value of α needs to be large enough to generate a convex function, but not too large as to overly underestimate the function. It is shown that:

$$
\begin{array}{rl}
\alpha \; \geq & \max \left\{ 0, \, -\dfrac{1}{2} \min\limits_{k, \, \mathbf{x}} \; \lambda_k(\mathbf{x}) \right\} \\
& \text{s.t.} \quad \mathbf{x}^L \; \leq \; \mathbf{x} \; \leq \; \mathbf{x}^U
\end{array}
\tag{26}
$$

where $\lambda_k(\mathbf{x})$ are the eigenvalues of the Hessian matrix of $NC(\mathbf{x})$. It is preferable to derive an analytical expression for the value of α using (26) as an equality. This will provide the tightest possible convex underestimation of $NC(\mathbf{x})$. In cases where this type of analysis is not possible, several methods have been developed which provide valid lower bounds on the eigenvalues of the Hessian matrix[22;23;24]. Transforming the equality constraints into two opposite signed inequalities, and substituting the underestimators, the complete lower bounding problem is then written as:

$$
\begin{aligned}
\min_{\hat{\mathbf{z}}_\mu, \boldsymbol{\xi}, \boldsymbol{\theta}} \quad & \sum_{m \in M} \sum_{\mu=1}^{r} \left(\hat{z}_{\mu,m} - \bar{z}_{\mu,m}\right)^2 \qquad (27) \\
\text{s.t.} \quad & \\
& \sum_{c=0}^{K} \xi_{e,c,j}\, \dot{\phi}_{e,c,j}(\tau_k) + \Delta\eta_e\, \mathcal{L}_{-g_j}(\xi_{e,k,j}, \boldsymbol{\theta}, t_{e,k}) \le 0 \\
& -\sum_{c=0}^{K} \xi_{e,c,j}\, \dot{\phi}_{e,c,j}(\tau_k) + \Delta\eta_e\, \mathcal{L}_{g_j}(\xi_{e,k,j}, \boldsymbol{\theta}, t_{e,k}) \le 0 \\
& \qquad e = 1, \ldots NE; \quad k = 1, \ldots, K; \quad j \in J \\
& \mathcal{L}_{\boldsymbol{h}}(\boldsymbol{\xi}_{e,k}, \boldsymbol{\theta}, t_{e,k}) \le \mathbf{0} \quad e = 1, \ldots, NE; \;\; k = 1, \ldots, K \\
& \mathcal{L}_{-\boldsymbol{h}}(\boldsymbol{\xi}_{e,k}, \boldsymbol{\theta}, t_{e,k}) \le \mathbf{0} \quad e = 1, \ldots, NE; \;\; k = 1, \ldots, K \\
& \xi_{e,0,j} - \sum_{c=0}^{K} \xi_{e-1,c,j}\, \phi_c(\tau = 1) = 0 \quad e = 2, \ldots, NE; \quad j \in J \\
& \xi_{1,0,j} - z_{0,j} = 0 \quad j \in J \\
& -\hat{z}_{\mu,m} + \sum_{c=0}^{K} \xi_{e_\mu,c,m} \phi_c(\tau_\mu) = 0 \quad m \in M; \;\; \mu = 1, \ldots, r \\
& \mathcal{L}_{\boldsymbol{f}}(\hat{\mathbf{z}}_\mu, \boldsymbol{\theta}) \le 0
\end{aligned}
$$

where $\mathcal{L}_{g_j}$, $\mathcal{L}_{-g_j}$, $\mathcal{L}_{\boldsymbol{h}}$, $\mathcal{L}_{-\boldsymbol{h}}$, and $\mathcal{L}_{\boldsymbol{f}}$ refer to the underestimating functions of the terms g_j, $-g_j$, $\boldsymbol{h}$, $-\boldsymbol{h}$, and $\boldsymbol{f}$ respectively. It should be noted that the relaxation also results in an increase in the degrees of freedom of the problem. The number of degrees of freedom is no longer equal to the size of the parameter set, but now includes the number of collocation coefficients which appear in the residual equations. This is a result of the necessary conversion of the nonlinear equality constraints into inequalities and the subsequent relaxation of those constraints.

4 Integration Approach

In this approach, there is no need to change the differential-algebraic system of equations and new variables are not introduced. The problem remains exactly as it is defined by (2).

4.1 Formulation

First, consider the differential-algebraic system of equations given in the formulation as a simple input-output map. The input is the values of the parameters, $\boldsymbol{\theta}$, while the output corresponds to the values of the states along time, $\mathbf{z}(t)$.

$$\boldsymbol{\theta} \longrightarrow \boxed{\begin{array}{rcl} \dot{\mathbf{z}}_j & = & \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t) \quad j \in J \\ \mathbf{0} & = & \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t) \\ \mathbf{z}_j(t_0) & = & \mathbf{z}_0 \quad t \in [t_0,t_f] \end{array}} \longrightarrow \mathbf{z}(t) \tag{28}$$

Pontryagin[36] presented two theorems concerning the continuity and differentiability of this map with respect to the parameters, $\boldsymbol{\theta}$. The theorems are derived and proven for a system of ODEs. It is possible to convert the DAE system in (28) into a set of ODEs either by explicitly solving $\mathbf{0} = \boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t)$ for the algebraic variables, $z_i \quad i \notin J$, and substituting into $\boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t)$ or through one differentiation of $\boldsymbol{h}(\mathbf{z},\boldsymbol{\theta},t)$ since the system is of order one[37].

Given the system

$$\dot{\mathbf{z}} = \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t) \quad , \tag{29}$$

the assumption that the right hand side of (29), $\boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t)$, and the partial derivatives,

$$\frac{\partial}{\partial \mathbf{z}} \, \boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t) \tag{30}$$

are defined and are continuous is some domain $\boldsymbol{\Gamma}$ of the space of variables t, $\mathbf{z}$, and $\boldsymbol{\theta}$, we have:

Theorem 1 (Pontryagin[36], page 170) *If $(t_0,\ \mathbf{z}_0,\ \boldsymbol{\theta}_0)$ is an arbitrary point of the domain $\boldsymbol{\Gamma}$, there exist positive numbers r and ρ such that for:*

$$|\boldsymbol{\theta} - \boldsymbol{\theta}_0| < \rho$$

the solution

$$\mathbf{z} \ = \ \boldsymbol{\psi}(t,\boldsymbol{\theta})$$

of (29) which satisfies the initial condition

$$\boldsymbol{\psi}(t_0,\boldsymbol{\theta}) = \mathbf{z}_0$$

is defined on the interval $|t - t_0| < r$ and is a continuous function of all the variables, t and $\boldsymbol{\theta}$, on which it depends.

Theorem 2 (Pontryagin[36], page 173) *Let the partial derivatives*

$$\boldsymbol{e}(\mathbf{z},\boldsymbol{\theta},t) = \frac{\partial}{\partial\boldsymbol{\theta}}\,\boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t)$$

exist and be continuous in the domain $\boldsymbol{\Gamma}$*. Let* $(t_0,\, \mathbf{z}_0,\, \boldsymbol{\theta}_0)$ *be some point of* $\boldsymbol{\Gamma}$*. Then, there exist positive numbers* r' *and* ρ' *such that for* $|t - t_0| < r'$, $|\boldsymbol{\theta} - \boldsymbol{\theta}_0| < \rho'$ *the solution* $\boldsymbol{\psi}(t,\boldsymbol{\theta})$ *of (29) which satisfies the initial condition:*

$$\boldsymbol{\psi}(t_0,\boldsymbol{\theta}) = \mathbf{z}_0\ ,$$

has continuous partial derivatives

$$\frac{\partial\boldsymbol{\psi}(t,\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\quad .$$

Corollary 1 (Pontryagin[36], page 177) *If all the partial derivatives of* $\boldsymbol{g}(\mathbf{z},\boldsymbol{\theta},t)$ *with respect to the variables* $\mathbf{z}$ *and* $\boldsymbol{\theta}$ *up to the* m^{th} *order inclusive exist and are continuous, then the functions* $\boldsymbol{\psi}(t,\boldsymbol{\theta})$ *also have continuous partial derivatives with respect to the parameters,* $\boldsymbol{\theta}$*, up to the* m^{th} *order inclusive.*

Given the conditions on the differential-algebraic system presented in section 2, and the aforementioned theorems, the values of the states, $\mathbf{z}$, at a given time t_μ can be defined as a set of twice continuously differentiable functions of the parameters, $\boldsymbol{\theta}$:

$$\mathbf{z}(t_\mu) \;=\; \boldsymbol{\mathcal{F}}(t_\mu,\boldsymbol{\theta})\quad . \tag{31}$$

Substituting this function into (2) results in:

$$\begin{aligned} &\min_{\boldsymbol{\theta},\,\hat{\mathbf{z}}_\mu}\ \sum_{m\in M}\sum_{\mu=1}^{r} (\hat{z}_{\mu,m} \;-\; \bar{z}_{\mu,m})^2 \\ &\text{s.t.} \\ &\qquad \hat{z}_{\mu,m} \;+\; \mathcal{F}_{m,\mu}(\boldsymbol{\theta}) \;=\; \mathbf{0} \qquad m\in M;\quad \mu = 1,\ldots,r \\ &\qquad \boldsymbol{f}(\hat{\mathbf{z}}_\mu,\boldsymbol{\theta}) \;\le\; 0 \end{aligned} \tag{32}$$

where $\mathcal{F}_{m,\mu}$ is the function which describes the value of state m at time point t_μ. The derivatives of this function are the values of the sensitivities of state m with respect to the parameters, $\boldsymbol{\theta}$, at the given time point t_μ. These sensitivities are determined by simultaneously integrating a set of linear equations with the initial system[37;38;39]. The set of equations, for this system, take the form of:

$$\begin{bmatrix} \mathbf{I}_{j\times j} & \mathbf{0}_{j\times k} \\ \mathbf{0}_{h\times j} & \mathbf{0}_{h\times k} \end{bmatrix} \left(\frac{\partial \dot{\mathbf{z}}}{\partial \boldsymbol{\theta}}\right) = \begin{bmatrix} \frac{\partial \boldsymbol{g}}{\partial \mathbf{z}} \\ \frac{\partial \boldsymbol{h}}{\partial \mathbf{z}} \end{bmatrix} \left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}\right) + \begin{bmatrix} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \end{bmatrix} \tag{33}$$

where j is the number of states with explicit derivatives, k are the number of states without explicit derivatives, and h is size of the vector of equations $\boldsymbol{h}$. Therefore the derivatives of the function $\mathcal{F}_{m,\mu}$ with respect to $\boldsymbol{\theta}$ are defined as:

$$\frac{\partial \mathcal{F}_{m,\mu}}{\partial \boldsymbol{\theta}} = \frac{\partial z_m}{\partial \boldsymbol{\theta}}(t_\mu) \ . \tag{34}$$

A numerical integration routine is used to generate the necessary function and gradient evaluations for the dynamic system in the course of the optimization algorithm[40].

The formulation (32) has the following properties:

1. There are no additional variables needed to deal with the dynamic nature of the model.
2. The possible error involved in the integration is small for well behaved systems.
3. Each function evaluation requires the integration of a differential-algebraic system of equations, therefore the computational expense in the determination of a local minimum increases as compared to the collocation approach.
4. There are currently no deterministic optimization methods which can guarantee the determination of the global minimum of this formulation.

4.2 Underestimating Formulation

In order to generate the underestimating formulation for this problem, the functions $\mathcal{F}_{m,\mu}(\boldsymbol{\theta})$ are treated as general nonconvex terms. These terms are underestimated by the addition of a quadratic term in θ_p multiplied by $\beta_{m,\mu}$ (note the similarity to the uniform diagonal shift approach for α's).

$$\mathcal{L}_{\mathcal{F}_{m,\mu}}(\boldsymbol{\theta}) = \mathcal{F}_{m,\mu}(\boldsymbol{\theta}) + \beta_{m,\mu} \sum_{p\in P} \left(\theta_p^U - \theta_p\right)\left(\theta_p^L - \theta_p\right) \tag{35}$$

where the β term is calculated in the same manner as the α term in the algebraic case. The functions $\boldsymbol{f}(\hat{\mathbf{z}}_\mu, \boldsymbol{\theta})$ are underestimated in the same way as for the collocation approach. The complete lower bounding problem is formulated as:

$$
\begin{aligned}
\min_{\boldsymbol{\theta},\, \hat{\mathbf{z}}_\mu} \quad & \sum_{m \in M} \sum_{\mu=1}^{r} \left(\hat{z}_{\mu,m} \;-\; \bar{z}_{\mu,m}\right)^2 & (36)\\
\text{s.t.} \quad & \\
& -\hat{z}_{\mu,m} \;+\; \mathcal{F}_{m,\mu}(\boldsymbol{\theta}) \;+\; \beta^{+}_{m,\mu} \sum_{p \in P} \left(\theta^U_p - \theta_p\right)\left(\theta^L_p - \theta_p\right) \le 0 \quad m \in M; \quad \mu = 1, \ldots, r \\
& \hat{z}_{\mu,m} \;-\; \mathcal{F}_{m,\mu}(\boldsymbol{\theta}) \;+\; \beta^{-}_{m,\mu} \sum_{p \in P} \left(\theta^U_p - \theta_p\right)\left(\theta^L_p - \theta_p\right) \le 0 \quad m \in M; \quad \mu = 1, \ldots, r \\
& \mathcal{L}_{\boldsymbol{f}}(\hat{\mathbf{z}}_\mu, \boldsymbol{\theta}) \;\le\; 0
\end{aligned}
$$

where P represents the full set of the variables $\boldsymbol{\theta}$. The values of the β's are determined from an expression identical to (26). In this case, the Hessian matrix of the function, $\mathcal{F}_{m,\mu}(\boldsymbol{\theta})$, is comprised of values for the second-order sensitivities of the state, z_m, with respect to the parameters, $\boldsymbol{\theta}$, at the given time point, t_μ:

$$
\mathcal{H}_{m,\mu} \;\equiv\; \frac{\partial^2 z_m}{\partial \boldsymbol{\theta}^2}(t_\mu) \quad . \tag{37}
$$

These sensitivities are determined by adding the set of equations given by (33) to the original problem, and hence generating a full augmented system. The system is then integrated with a sensitivity analysis, thus generating the sensitivities of (33) with respect to the parameters (an alternative method is presented by Vassiliadis et al.[41]). The value of the β parameters are determined by:

$$
\begin{aligned}
\beta_{\mu,i} \;&\ge\; -\frac{1}{2} \min_{\boldsymbol{\theta}} \lambda^{min}_{\mu,i}(\boldsymbol{\theta}) & (38)\\
\beta_{\mu,i} \;&\ge\; 0
\end{aligned}
$$

where $\lambda^{min}_{\mu,i}$ is the minimum eigenvalue of the matrix $\mathcal{H}_{\mu,i}$. The elements of $\mathcal{H}_{\mu,i}$ can not be written as algebraic functions of $\boldsymbol{\theta}$. Therefore, the minimization given in (38) can not be determined exactly. The elements of the matrix, however, can be calculated through an integration of the augmented system at fixed values of $\boldsymbol{\theta}$. As a result, three different methods are used to calculate the β values:

Method 1: Constant values

In this approach, the values which will be used for the β parameters are preselected. The drawback is that no second order information, which could be made available, is being used and the validity of the underestimator is not known.

Method 2: Sampling of the Hessian Matrix

The values of the elements of $\mathcal{H}$ can be calculated at selected values of the parameters, θ. Therefore, a number of points are selected to evaluate the Hessian matrix and calculate the eigenvalues of these matrices. The minimum of theses eigenvalues is then used to determine the value of β. The number of points sampled and the sampling method are both parameters in this approach. In all cases a uniformly random sample is generated. The number of points used in the sample has both an effect on the validity of the underestimator and the time required to generate it. The more points used, the better the approximation becomes, but the time required increases. The size of the sampled set needs to be large enough to generate a valid value for β, but not too large so as to require a substantial computational expense.

Method 3: Sampling with Interval Hessian Calculation

In this approach, the values of each element of $\mathcal{H}$ are also determined at selected values of the parameters, but the eigenvalues of these Hessian matrices are not directly evaluated. Instead, an interval Hessian matrix is generated from the minimum and maximum of every element over the sampled set. A valid lower bound on the minimum eigenvalue of this matrix can then by determined using methods presented by Adjiman and Floudas[22], and Adjiman et al.[23;24]. It has been shown that for algebraic problems the value of α calculated using interval methods, overestimates the value needed for convexity. Therefore, fewer sampled points can be used to reasonably insure a convex lower bounding problem.

5 Algorithmic Procedure

A step by step presentation of the algorithm as implemented is given below.

Step 1. Initialize the problem

1. Set the absolute, ϵ^{abs}, or relative, ϵ^{rel}, convergence tolerance.

2. Set the iteration counter, *iter*, equal to zero.

3. Set reasonably tight bounds on the fitted data variables, $\hat{\mathbf{z}}_\mu$, from statistical considerations, or prior knowledge.

4. Set reasonable and loose bounds on the parameters, $\boldsymbol{\theta}$, and the collocation coefficients, $\boldsymbol{\xi}$.

Step 2. Tighten initial variable bounds

1. Update the bounds on the parameters, $\boldsymbol{\theta}$, and the collocation coefficients, $\boldsymbol{\xi}$, by solving a series of feasibility problems based on the underestimating formulation.

$$x_j^{l,*}/x_j^{u,*} = \left\{ \begin{array}{rl} \min\limits_{\boldsymbol{\theta},\hat{\mathbf{z}}_\mu,\boldsymbol{\xi}} / \max\limits_{\boldsymbol{\theta},\hat{\mathbf{z}}_\mu,\boldsymbol{\xi}} & x_j \\ \text{s. t.} & \mathcal{L}(\boldsymbol{\theta},\hat{\mathbf{z}}_\mu,\boldsymbol{\xi}) \leq \mathbf{0} \\ & \hat{\mathbf{z}}_\mu^L \leq \hat{\mathbf{z}}_\mu \leq \hat{\mathbf{z}}_\mu^U \\ & \boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U \\ & \boldsymbol{\xi}^L \leq \boldsymbol{\xi} \leq \boldsymbol{\xi}^U \end{array} \right. \tag{39}$$

where $\mathbf{x}$ represents the vector $[\boldsymbol{\theta}, \boldsymbol{\xi}]$ and $\mathcal{L}(\boldsymbol{\theta},\hat{\mathbf{z}}_\mu,\boldsymbol{\xi})$ represents the set of all constraints which appear in the underestimating problem.

2. Calculate the relative change of all the variable bounds:

$$\Delta_B = \sum_{i \in I} \left(1 - \frac{x_i^{new,u} - x_i^{new,l}}{x_i^{old,u} - x_i^{old,l}} \right) \tag{40}$$

   where the superscript *old* refers to the bounds on the variables before (39) is solved, and *new* refers to the bounds afterwards.

3. If $\Delta_B < 0.1$ or a given number of bounds iterations has been reached, then proceed to Step 3, otherwise go back to 1. using the new bounds on the variables.

Step 3. Determine Initial Upper and Lower Bounds on the Global Solution

1. Using the bounds on the variables determined in Step 2, solve the convex relaxation to local optimality to generate a lower bound (LB) on the global solution.

2. Save the solution and all the variable values

$$obj^* \rightarrow obj^{1,1} \qquad \mathbf{x}^* \rightarrow \mathbf{x}^{1,1} \quad .$$

3. Using the lower bounding solution as a starting point, solve the original nonconvex problem to local optimality to determine an upper bound (UB) on the global solution.

4. Save the solution and all the variable values

$$obj^* \rightarrow UB \qquad \mathbf{x}^* \rightarrow \mathbf{x}^{UB} \quad .$$

Step 4. Check for convergence
if UB - LB $\leq \epsilon^{abs}$ or (UB - LB)/UB $\leq \epsilon^{rel}$ then terminate with the global solution:

$$obj^{glo} = obj^{UB} \qquad \mathbf{x}^{glo} = \mathbf{x}^{UB}$$

otherwise $iter = iter + 1$.

Step 5. Update variable bounds

1. Select a given number of collocation variables, $\boldsymbol{\xi}$, to have their bounds updated.

2. Solve (39) for each of the selected variables.

Step 6. Branch the region

1. Determine which variable from the branching set that causes the greatest deviation between the original problem and the convex relaxation. The branching set may contain all or some of the parameter variables and the collocation coefficients. The contribution is calculated by summing a term deviation for a given variable over all the terms that variable participates in. The term deviation is defined by:

$$\delta_i \; = \; f(\mathbf{x}^{sol}) \; - \; \mathcal{L}_f(\mathbf{x}^{sol}) \tag{41}$$

where $f(\mathbf{x})$ is the original nonconvex term, $\mathcal{L}_f(\mathbf{x})$ is the convex relaxation, and $\mathbf{x}^{sol}$ are the values of variables at the solution to the convex relation in the current region. As an example, consider an α or β underestimated term. The deviation for a given variable x_j and term i is calculated as:

$$\delta_i^j \; = \; \alpha_i(x_j^U \; - \; x_j^{sol})\,(x_j^L \; - x_j^{sol}) \tag{42}$$

2. Branch the region into two $(r = 1, 2)$ by bisecting on the selected variable.

Step 7. Solve the Upper and Lower Problems in each new region

1. Solve the convex relaxation from a randomly chosen starting point.
2. Save the solution if it is less than the current upper bound (UB)

$$\text{if } obj^{lower} < UB \qquad \text{then} \qquad obj^{lower} \rightarrow obj^{iter,r} \qquad \mathbf{x}^* \rightarrow \mathbf{x}^{iter,r} \quad .$$

   Otherwise, if the solution is greater than the upper bound, or if there is no feasible solution, remove the region from consideration and go to Step 8.

3. Solve the upper problem in this region using the solution of the lower problem as a starting point.
4. If the solution is less than the current upper bound, then update the bound

$$\text{if } obj^{upper} < UB \qquad \text{then} \qquad obj^{upper} \rightarrow UB \qquad \mathbf{x}^* \rightarrow \mathbf{x}^{UB} \quad .$$

Step 8. Update Lower Bound

1. Select the next region to be explored as the one with the lowest solution to the relaxed problem.
2. The region is removed from the list of store regions, and the solution becomes the new lower bound

$$LB = \min_{iter,r} \;\; obj^{iter,r} \quad .$$

3. Return to Step 4.

Differences between Formulations: There are some differences in the algorithm used for the two different approaches. These are:

- In the integration approach, no collocation variables are present, so initially only the bounds on the parameters variables are updated.

- There are no bounds updates performed at each iteration for the integration approach since no collocation variables exist.

Remark 3 Note that in the collocation based approach the proposed method offers a theoretical guarantee of attaining the global solution of the transformed algebraic model, (15). In the integration based approach, the proposed method offers a theoretical guarantee of attaining the global solution of the original problem (2) as long as rigorous values of the β parameters or rigorous bounds on the β parameters are obtained.

6 Implementation

The algorithmic procedure presented in section 5 has been implemented in an extensive C program with a intuitive front end parser. All the necessary differentiations and generation of additional constraints are performed automatically by the parser. A link to the MINOPT optimization program[40] is used in order to perform the local optimizations and integrations. MINOPT itself has links to various local solvers, including SNOPT[42], MINOS[43] and NPSOL [44], as well as the integration routines DASOLV[45] and DAESSA[40]. MINOS is used for the local solution of the algebraic problems encountered in the collocation approach, while SNOPT and DASOLV are used for the integration approach. Figure 2 shows the program flow among these different parts. A sample input file can be found in Appendix C.

7 Computational Studies

In order to illustrate the theoretical and computational aspects of both proposed approaches, six example problems dealing with the estimation of kinetic parameters from time series data are presented. The performance of each approach and comparisons between the two approaches will be discussed. All example problems were solved on an HP J2240, using a single processor. The data used in each example can be found in Appendix A, and the explicit formulations of all the case studies are provided in Floudas et al.[46].

7.1 First-order Irreversible Series Reaction

This model represents a first order irreversible chain reaction as presented by Tjoa and Biegler[18].

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

Only the concentrations of components A and B were measured, therefore component C does not appear in the model used for estimation. The differential equation model takes the form:

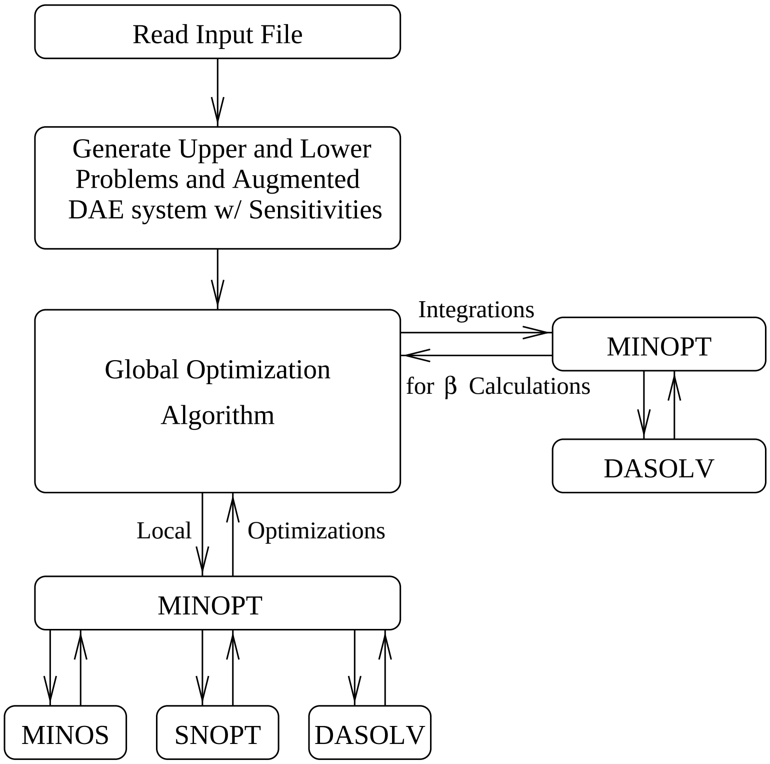

Read Input File
Generate Upper and Lower Problems and Augmented DAE system w/ Sensitivities
Global Optimization Algorithm
Integrations
for β Calculations
MINOPT
DASOLV
Local
Optimizations
MINOPT
MINOS
SNOPT
DASOLV

Figure 2: Program Flow

$$
\begin{aligned}
\frac{dz_1}{dt} &= -\theta_1 z_1 \\
\frac{dz_2}{dt} &= \theta_1 z_1 - \theta_2 z_2 \\
\mathbf{z}_0 &= [1,\ 0] \qquad t \ \in \ [0,\ 1]
\end{aligned}
\tag{43}
$$

where the state vector, $\mathbf{z}$, is defined as $[A,\ B]$, and the parameter vector, $\boldsymbol{\theta}$, is defined to be $[k_1,\ k_2]$. The data used in the study was generated with values for the parameters of $\boldsymbol{\theta} = [5, 1]$ with no added error.

Collocation Approach

The state profiles are approximated using different orders (K) of polynomials on 5 equal length finite elements. The complete formulation of this problem is given below.

$$
\min_{\hat{\mathbf{z}}_\mu,\, \boldsymbol{\xi},\, \boldsymbol{\theta}} \quad \sum_{m=1}^{2} \sum_{\mu=1}^{10} \left(\hat{z}_{\mu,m} \ - \ \bar{z}_{\mu,m}\right)^2 \tag{44}
$$

s.t.

$$
\begin{aligned}
&\sum_{c=0}^{K} \xi_{e,c,1}\, \dot{\phi}_{e,c,1}(\tau_k) \ - \ \Delta\eta_e(-\theta_1\, \xi_{e,k,1}) \ = \ 0 \quad e = 1, \ldots, 5; \ \ k = 1, \ldots, K \\
&\sum_{c=0}^{K} \xi_{e,c,2}\, \dot{\phi}_{e,c,2}(\tau_k) \ - \ \Delta\eta_e(\theta_1\, \xi_{e,k,1} - \theta_2\, \xi_{e,k,2}) \ = \ 0 \quad e = 1, \ldots, 5; \ \ k = 1, \ldots, K \\
&\xi_{e,0,j} \ - \ \sum_{c=0}^{K} \xi_{e-1,c,j}\, \phi_c(\tau = 1) \ = \ 0 \qquad e = 2, \ldots, 5; \qquad j = 1, \ldots, 2 \\
&\xi_{1,0,j} \ - \ z_{0,j} \ = \ 0 \quad j = 1, \ldots 2 \\
&-\hat{z}_{\mu,m} \ + \ \sum_{c=0}^{K} \xi_{e_\mu,c,m} \phi_c(\tau_\mu) \ = \ 0 \quad m = 1, \ldots, 2; \ \ \mu = 1, \ldots 10 \\
&\mathbf{z}_0 \ = \ (1, 0) \qquad t \in [0, 1] \\
&t_\mu = (0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1) \\
&\Delta\eta_e = (0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2)
\end{aligned}
$$

The nonconvexities in (44) are the bilinear terms, $\pm\theta_1\, \xi_{e,k,1}$ and $\theta_2\, \xi_{e,k,2}$ in the residual equations. These terms are relaxed using the linear cuts defined by (23).

The bounds on the variables were initially set to $\boldsymbol{\theta} \in [0, 10]$, $\boldsymbol{\xi} \in [0, 1]$, and $\hat{\mathbf{z}}_\mu \in [\bar{\mathbf{z}}_\mu \pm 0.1]$. The bounds on the parameters, $\boldsymbol{\theta}$, and the collocation coefficients, $\boldsymbol{\xi}$, were initially updated once. Table 1 shows the global solutions, the number of iterations, time required to reach the solution, and the number of nonlinear variables in the formulation, for polynomials of order 3, 4, and 5. In these runs, no variable bounds were updated at each iteration, branching was performed only on the parameters, and a relative convergence tolerance of 1% was used. As the order of approximation is increased, the mean squared error decreases and the original values of the parameters are obtained. At the same time the number of nonlinear variables also increases. This change not only results in a slight increase in the number of iterations required, but also, in a large increase in the solution time required. A mere doubling of the number of collocation coefficients results in a five-fold increase in the computational effort. This increase is related to the way in which the local solver scales with the number of nonlinear variables.

Order(K)	Variables	Iterations	CPU sec.	θ_1	θ_2	Obj
3	22	34	2.92	5.0017	0.9995	6.276×10^{-6}
4	32	36	5.89	5.0016	1.0001	1.289×10^{-6}
5	42	38	10.77	5.0035	1.0000	1.185×10^{-6}

Table 1: Solutions for example 1 with different polynomial approximations on 5 finite elements.

Figure 3 plots the number of iterations and the solution time required for a 5th order polynomial approximation versus the number of variables whose bounds are updated at each iteration. The selection of the variables to update is done randomly from the set of nonlinear $\boldsymbol{\xi}$ variables. This random selection results in a slight variation in the number of iterations required from run to run, therefore the results are the average of three runs. As the number of updated variables increases, the iterations required to solve the problem decrease, as is expected. At the same time, even though less iterations are required, the overall solution time increases. Since this is a relatively small problem, the decrease in the number of iterations, is not large enough to offset the additional time required to solve the bounds updating problems.

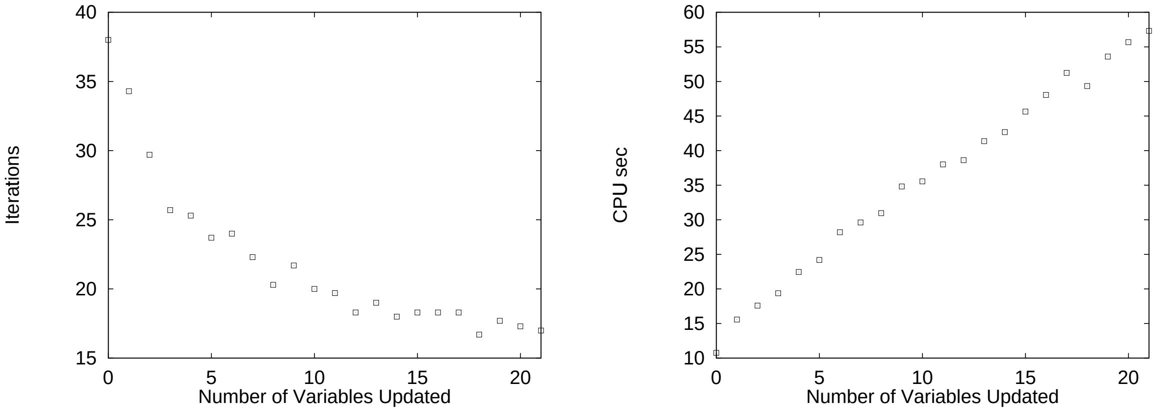


Figure 3: Solution times using 5^{th} order polynomials versus variable bounds updating for example 1.

Integration Approach

The formulation for this approach is given below:

$$
\begin{aligned}
&\min_{\hat{\mathbf{z}}_\mu,\boldsymbol{\theta}} \quad \sum_{m=1}^{2}\sum_{\mu=1}^{10}\left(\hat{z}_{\mu,m} \; - \; \bar{z}_{\mu,m}\right)^2 \qquad (45)\\
&\text{s.t.}\\
&\boxed{\begin{aligned}\dot{z}_1 &= -\theta_1\, z_1\\ \dot{z}_2 &= \theta_1\, z_1 - \theta_2\, z_2\\ \mathbf{z}(t_0) &= (1,\,0) \quad t \,\in\, [0,1]\end{aligned}} \;\longrightarrow\; \mathcal{F}_{m,\mu}(\boldsymbol{\theta})\\
&\hat{z}_{\mu,m} \;+\; \mathcal{F}_{m,\mu}(\boldsymbol{\theta}) \;=\; 0 \quad m = 1,2\;;\quad \mu = 1,\ldots,10
\end{aligned}
$$

The integration approach converges to a solution of 1.1858×10^{-6} with parameter values of $\boldsymbol{\theta} = [5.0035,\ 1.0000]$. The bounds on the parameters and the fitted data variables are initially set to the same values used in the collocation approach. Initial bounds updating is performed on the parameter variables until the convergence criteria given by (40) is reached. A relative convergence tolerance, ϵ^{rel}, of 1 % is used. Figure 4 shows the number of iterations and effort required to obtain the global solution using constant β values.

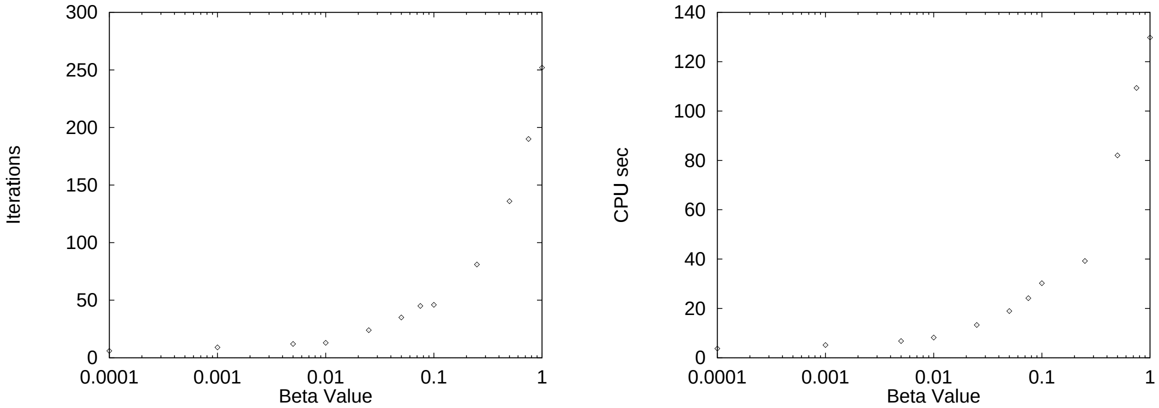


Figure 4: Solution times using integration versus the value of β for Example 1. The same β value is used for all the constraints.

By evaluating the eigenvalues of the Hessian matrices at 5000 randomly generated points in the parameter space, the values of β were found to range from 0 to 0.5 with an average value of 0.123. The evaluation of the Hessian matrices is expensive, in this rather small case (2 states, 2 parameters), approximately 70 CPU seconds are required. The problem was solved using these β values initially and either keeping them constant, or updating them in each region. In each region, at least 20 sampled points are used to determine the β value. In addition, the initial bounds updating on the parameter values was studied. The bounds were either updated once (as in the collocation formulation) or until the convergence criteria given by (40) was reached. The results are shown in Table 2. In all cases, initially updating the bounds on the parameter variables multiple times, not only reduces the number of iterations required, but also the computational time.

In this case, unlike in the collocation approach, only two variables are being updated, and therefore the computational requirement is small compared to the reduction in iterations.

β Values	$\boldsymbol{\theta}$ initial updates	iterations	CPU sec.
constant	once	35	20.05
constant	convergence	20	13.30
updated	once	27	19.05
updated	convergence	14	13.82

Table 2: Results of Integration Formulation for Example 1.

Also, by updating the β values, the number of iterations required is reduced, but the computational expense remains relatively the same. Updating these values requires expensive additional Hessian evaluations to ensure that at least 20 points are used in each region.

7.2 First-order Reversible Series Reaction

This model represents the same series reaction given in example 1, but the reactions are reversible. The model appears in Tjoa and Biegler[18].

$$A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B \underset{k_4}{\overset{k_3}{\rightleftharpoons}} C$$

In this case all the components are measured and therefore their concentrations are included in the model used for estimation. The differential equation model takes the form:

$$\begin{aligned} \frac{dz_1}{dt} &= -\theta_1 z_1 + \theta_2 z_2 \\ \frac{dz_2}{dt} &= \theta_1 z_1 - (\theta_2 + \theta_3) z_2 + \theta_4 z_3 \\ \frac{dz_3}{dt} &= \theta_2 z_2 - \theta_4 z_3 \\ \mathbf{z}_0 &= [1,\ 0,\ 0] \qquad t \ \in \ [0,\ 1] \end{aligned} \tag{46}$$

where the state vector, $\mathbf{z}$, is defined as $[A,\ B,\ C]$, and the parameter vector, $\boldsymbol{\theta}$, is defined to be $[k_1,\ k_2,\ k_3,\ k_4]$. Two different data sets were generated using the following values of the parameters, $\boldsymbol{\theta} = [4,\ 2,\ 40,\ 20]$. One set was used without any error being added, and another with a small amount of random error added.

Collocation Approach

The state profiles are approximated using 5th order polynomials on 6 finite elements with the element boundaries at $t = [0,\ 0.1,\ 0.2,\ 0.3,\ 0.45,\ 0.7]$. All of the nonlinearities in this formulation are in the form of bilinearities. Using the error free data, the solution obtained has an objective value of 3.367×10^{-7} with parameter values of $\boldsymbol{\theta} = [4.001,\ 2.001,\ 39.80,\ 19.90]$. The initial bounds on the variables were set to: $\theta_{1,2} \in [0, 10]$, $\theta_{3,4} \in [10, 50]$, $\boldsymbol{\xi} \in [0, 1]$, and the initial bounds on the fitted data variables were varied. Table 3 shows the results using different bounds for these variables. The solutions were obtained by initially updating the variable bounds three times, branching on the parameters, $\boldsymbol{\theta}$, using an absolute convergence tolerance of 1×10^{-8}, and updating no variables at each iteration.

Bounds on $\hat{\mathbf{z}}_\mu$	Iterations	CPU sec.
$\bar{\mathbf{z}}_\mu \pm 0.05$	338	549.07
$\bar{\mathbf{z}}_\mu \pm 0.03$	189	398.66
$\bar{\mathbf{z}}_\mu \pm 0.01$	56	272.91

Table 3: Results using collocation approach with error free data for Example 2.

Using the error added data with the same 5th order polynomial approximation on 6 elements, the algorithm converged to a solution with an objective value of 1.586×10^{-3} and parameter values of $\boldsymbol{\theta} = [4.021, 2.052, 39.45, 19.62]$. The same bounds on the parameters and collocation coefficients were used, and variable bounds update and branching were performed in the same manner. Table 4 shows the results using different bounds on the fitted data variables, with a relative convergence tolerance of 1%.

Bounds on $\hat{\mathbf{z}}_\mu$	Iterations	CPU sec.
$\bar{\mathbf{z}}_\mu \pm 0.050$	1440	1613.33
$\bar{\mathbf{z}}_\mu \pm 0.025$	349	568.44

Table 4: Results using collocation approach with error added data for Example 2.

It is interesting to note how much additional effort is required to solve the problem with error in the data as opposed to error free data. Also, the collocation approach is very sensitive to the initial bounds chosen for the fitted data variables. Even though these variables do not directly participate in any nonconvex terms, the bounds chosen have a large indirect effect. The initial bounds on the collocation coefficients, which do appear in non-convex terms, are derived from the fitted data variables though the initial bounds updating procedure. Tighter initial bounds on $\hat{\mathbf{z}}_\mu$ result in tighter bounds on $\boldsymbol{\xi}$.

Integration Approach

The integration approach has one additional property that has not been discussed. Due to the relaxation used in developing the underestimating formulation, it is possible to violate the mass balance at the solution of the lower bounding problem. Therefore it is advantageous to add the algebraic mass balance constraints into the lower bounding problem (similar to the approach used by Quesada and Grossmann[47]). These constraints take the form:

$$\sum_{m=1}^{3} \hat{z}_{\mu,m} \;=\; 1 \qquad \mu = 1, \ldots, 20 \quad . \tag{47}$$

Since (47) is linear in the fitted data variables, it is not necessary to underestimate this constraint.

Using the error free data, the approach converged to a solution with an objective value of 1.890×10^{-7} with parameter values of $\boldsymbol{\theta} = [4.000, 2.000, 40.01, 20.01]$. The problem was solved using different bounds on the fitted data variables with the mass balance constraint given by (47) and using a sampling method to calculate the β values (25 points used initially and at least 10 in each region). As in the collocation approach, the initial bounds on the parameters were set to $\theta_{1,2} \in [0, 10]$, $\theta_{3,4} \in [10, 50]$, initial bounds updates were performed on the parameters until convergence was reached, and branching was on the parameters. The results appear in Table 5.

Bounds on $\hat{\mathbf{z}}_\mu$	Iterations	CPU sec.
$\bar{\mathbf{z}}_\mu \pm 0.05$	280	1546.05
$\bar{\mathbf{z}}_\mu \pm 0.03$	284	1646.34
$\bar{\mathbf{z}}_\mu \pm 0.01$	289	1899.82

Table 5: Results using the integration approach with error free data for Example 2.

Using the error added data, the formulation converged to a solution with an objective value of 1.587×10^{-3} with parameter values of $\boldsymbol{\theta} = [4.020, 2.052, 39.65, 19.72]$. The results are obtained using the same bounds, updating procedure, and branching as for the error free data. In this case, a relative tolerance of 1% is used for convergence. The results using different bounds on the fitted data variables are shown in Table 6. As an illustration of the difference made by adding the mass balance constraint, using bounds for $\hat{\mathbf{z}}_\mu$ of $\bar{\mathbf{z}}_\mu \pm 0.050$ without the mass balance constraint, the solution requires 2561 iterations and 7604.83 CPU sec. (a 25% increase in computational expense).

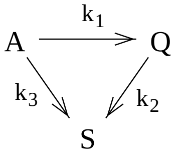
Bounds on $\hat{\mathbf{z}}_\mu$	Iterations	CPU sec.
$\bar{\mathbf{z}}_\mu \pm 0.050$	2105	6017.6
$\bar{\mathbf{z}}_\mu \pm 0.025$	2073	6164.3

Table 6: Results using the integration approach with error added data for Example 2.

From the results for both the error free and the error added data, it is clear that the convergence characteristics of the integration approach are not dependent on the bounds chosen for the fitted data variables. For this example the integration formulation requires more iterations and a greater computational effort.

7.3 Catalytic Cracking of Gas Oil

This model represents the catalytic cracking of gas oil (A) to gasoline (Q) and other side products (S).



Only the concentrations of A and Q were measured, therefore the concentration of S does not appear in the model for estimation. This model was studied by Tjoa and Biegler[18]. This reaction scheme involves nonlinear reaction kinetics, rather than the simple first order kinetics in the previous two examples. The differential equation model takes the form:

$$\begin{aligned} \frac{dz_1}{dt} &= -(\theta_1 + \theta_3) z_1^2 \\ \frac{dz_2}{dt} &= \theta_1 z_1^2 - \theta_2 z_2 \\ \mathbf{z}_0 &= [1,\ 0] \qquad t \ \in \ [0,\ 0.95] \end{aligned} \tag{48}$$

where the state vector, $\mathbf{z}$, is defined as $[A,\ Q]$, and the parameter vector, $\boldsymbol{\theta}$, is defined to be $[k_1,\ k_2,\ k_3]$. The data used in this study was generated using values for the parameters of $\boldsymbol{\theta} = [12,\ 8,\ 2]$ with a small amount of random error added.

Collocation Approach

For this example, we consider 6th order polynomials on 3 finite elements with the element boundaries at $t = [0,\ 0.1,\ 0.3]$. The model defined by (48) is nonlinear in the states. To generate the convex relaxation, the substitution, $b = z_1^2$, is added into the formulation.

$$\begin{aligned} \frac{dz_1}{dt} &= -\theta_1\, b + \theta_3\, b \\ \frac{dz_2}{dt} &= \theta_1\, b - \theta_2 z_2 \\ 0 &\le b \ - \ z_1^2 \\ 0 &\le -b \ + \ z_1^2 \end{aligned} \tag{49}$$

In (49) the nonconvexities take the form of bilinear terms and univariate concave terms $(-z_1^2)$. The underestimating problem then includes the convex hull of each of these terms. The initial bounds on the variables were set to $\boldsymbol{\theta} \in [0,\ 20]$, $\hat{\mathbf{z}}_\mu \in [\bar{\mathbf{z}}_\mu \pm 0.05]$, and $\boldsymbol{\xi} \in [0,\ 1]$. The bounds on the substitution variable, b, were calculated from the bounds on $\boldsymbol{\xi}$ in each region. Initial bounds updating was performed on the parameters, $\boldsymbol{\theta}$, and the collocation coefficients, $\boldsymbol{\xi}$. The approach converges to a solution with an objective value of 2.6384×10^{-3} with parameter values of $\boldsymbol{\theta} = [12.212,\ 7.980,\ 2.222]$. Table 7 shows results obtained by branching on only the parameters while varying the number of $\boldsymbol{\xi}$ variables whose bounds are updated at each iteration. Table 8 shows results obtained by branching on both the parameters and the collocation coefficients.

For this example it is necessary to update a number of variable bounds at each iteration and branch on the collocation coefficients in addition to the parameter variables to achieve

	1 % conv.		0.1 % conv.		0.05 % conv.	
Bounds Updated	Iter	CPU sec.	Iter	CPU sec.	Iter	CPU sec.
0	-	-	-	-	-	(1.70%)
2	134	146.46	-	-	-	(0.99%)
5	31	102.61	-	-	-	(0.27%)
10	28	129.12	49	178.32	73	233.42
15	26	158.35	42	211.25	48	229.46

Table 7: Results using the collocation approach with branching on the parameters only for Example 3. - does not achieve convergence in 1000 iter, (x%) reaches a relative difference of x % in 1000 iterations.

	1 % conv.		0.1 % conv.		0.05 % conv.	
Bounds Updated	Iter	CPU sec.	Iter	CPU sec.	Iter	CPU sec.
0	233	150.97	741	315.24	926	361.71
2	47	96.35	130	157.41	157	175.02
5	34	106.92	71	155.89	79	166.34
10	28	129.57	51	182.59	68	221.34
15	24	147.23	47	224.70	51	237.92

Table 8: Results using the collocation approach with branching on the parameters and the collocation coefficients for Example 3.

convergence. This is related to the nonconvexities present in this model. Since the parameters do not appear in every nonconvex term, branching only on these variables will not allow a complete tightening of the relaxation. Therefore it is necessary to either update a large number of the collocation coefficients, or add these variables to the branching set in order to achieve a tighter convergence tolerance.

Integration Approach

Using the same initial bounds on the parameters and fitted data variables, the approach converges to a solution with an objective value of 2.6557×10^{-3} and parameter values of $\boldsymbol{\theta} = [12.214,\ 7.980,\ 2.222]$. The initial bounds on the parameters were updated until convergence, and branching was performed only on these variables. Table 9 shows the results using the three different methods for calculating β values. The constant β values were determined by sampling 5000 points. The sampling 1 method used 25 points initially and at least 10 points in each region, with sampling 2 used 50 points initially and at least 20 points at each region. The interval 1 method generated interval Hessian matrices using 25 points initially and at least 10 points in each region, while interval 2 used 50 points initially and at least 20 points in each region. The eigenvalues of the interval Hessian matrices are calculated using the Gerschgorin method[23;24].

	1 % conv.		0.1 % conv.		0.05 % conv.	
β Calculation	Iter	CPU sec.	Iter	CPU sec.	Iter	CPU sec.
Constant	999	1185	-	-	-	-
Sample 1	40	79.77	56	100.21	60	104.86
Sample 2	46	103.11	65	133.17	70	140.61
Interval 1	60	106.50	81	137.30	87	141.32
Interval 2	58	112.92	74	147.67	80	156.53

Table 9: Results using integration approach for Example 3, (-) did not reach that convergence level in 1000 iterations.

The results clearly show that updating the β values at each iteration is necessary to achieve convergence in a reasonable amount of time. The use of the interval based methods for calculating β values required more iterations to converge to the solution. This is due to the fact that these interval methods can overestimate the value of β required to achieve convexity[22]. The difference within each method can be simply explained as a result of a variation in the values of β from using different randomly sampled sets of points in the calculation. The integration approach performs better at higher convergence levels than the collocation approach. This is directly related to the existence of the collocation variables, $\boldsymbol{\xi}$, discussed earlier.

7.4 Bellman's Problem

This problem describes a reversible homogeneous gas phase reaction of the form:

$$2NO + O_2 \rightleftharpoons 2NO_2$$

This model appeared first in Bellman et al.[4], and was also studied Varah[48] and Tjoa and Biegler[18]. The resulting differential equation is the result of many normalizations for the given experimental data and is nonlinear in the states,

$$\begin{aligned} \frac{dz}{dt} &= p_1(c_1 - z)(c_2 - z)^2 - p_2 z^2 \\ z_0 &= 0 \qquad t \in [0,\ 39.0] \end{aligned} \tag{50}$$

with $\mathbf{c} = [126.2,\ 91.9]$. The parameters, p_1 and p_2, in (50) are on the order of 1×10^{-6} and 1×10^{-4}, while the state, z, varies from 0 to 45. This results in a very difficult problem to solve, even from a local standpoint. Therefore, an exponential transformation for the parameters is used:

$$\theta_i = -\ln p_i \qquad i = 1,\ 2 \quad . \tag{51}$$

The reformulated model becomes:

$$\begin{aligned} \frac{dz}{dt} &= e^{-\theta_1}(c_1 - z)(c_2 - z)^2 - e^{-\theta_2} z^2 \\ z_0 &= 0 \qquad t \in [0,\ 39.0] \quad . \end{aligned} \tag{52}$$

Collocation Approach

The development of an underestimator for the model (52) can be accomplished by two different approaches. First, the two terms,

$$e^{-\theta_1}(c_1 - z)(c_2 - z)^2 \quad \text{and} \quad e^{-\theta_2} z^2$$

are underestimated using an α based approach. Analytical expressions for the values of α were determined using an eigenvalue analysis of the Hessian matrices and can be found in Appendix B. Therefore, the values used are the least possible to ensure convexity over the domain of interest. A second approach, is to expand the model and underestimate simpler terms. This results in:

$$\frac{dz}{dt} = -e^{-\theta_1} z^3 + (c_1 + 2\,c_2)\ e^{-\theta_1} z^2 - \left(c_2^2 + 2\,c_1\,c_2\right) e^{-\theta_1} z + \left(c_1\,c_2^2\right)\ e^{-\theta_1} - e^{-\theta_2} z^2 \quad . \tag{53}$$

The nonconvex terms in (53) fall under two different categories. First, the terms $\pm e^{-\theta_i}\, z^n$ are treated as general nonconvex terms. For each of these terms, analytical expressions for the value of α were also determined using an eigenvalue analysis of their Hessian matrices. Secondly, the term $-e^{-\theta_1}$, is treated as a univariate concave term and underestimated with a line segment.

The state profiles were approximated using 3^{rd} order polynomials on 3 finite elements, with element boundaries at $t = [0,\ 5,\ 20]$. A solution with an objective function value of 22.5684, and transformed parameter values of $\boldsymbol{\theta} = [12.29098,\ 8.17059]$ was found. These values of $\boldsymbol{\theta}$ give original parameter values of $p_1 = 4.5930 \times 10^{-6}$, and $p_2 = 2.8285 \times 10^{-4}$. A global run was performed using initial bounds updating on the $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ variables until convergence, and updating on all $\boldsymbol{\xi}$ variables at every iteration. Branching was performed not only on the parameter variables, but also on the collocation coefficients. Each of the two different underestimation methods was used with different bounds on $\boldsymbol{\theta}$ and the fitted data variables, $\hat{z}_\mu$. Table 10 shows the results for these runs. Loose initial bounds refers to bounds of $\boldsymbol{\theta} \in [5, 15]$ and $\hat{z}_\mu \in [\bar{z}_\mu \pm 5]$, while tight initial bounds refers to $\theta_1 \in [10, 14]$, $\theta_2 \in [6, 10]$ and $\hat{z}_\mu \in [\bar{z}_\mu \pm 1]$ for most μ, but expanded were necessary.

Formulation	Initial Bounds	Iterations	CPU sec.	Relative Conv.
original	loose	10000	9,708	75.0 %
original	tight	9028	7,835	0.1 %
expanded	loose	10000	12,222	78.9 %
expanded	tight	10000	4,939	9.0 %

Table 10: Global optimization solution times for Bellman's problem using the collocation approach. Original formulation given by (52) and the expanded formulation given by (53).

Only the original formulation with very tight initial bounds was able to reach 0.1% convergence in less than 10,000 iterations. The inability to achieve convergence can be linked to the form and number of nonconvexities in the problem. In each formulation, the values of α needed ranged in magnitude from 100 to 1×10^8, and were insensitive to region size. Therefore, even within small regions, the gap between the underestimator and the original problem is large leading to poor convergence.

Integration Approach

Using the integration approach with expanded bounds on the transformed parameters $\boldsymbol{\theta} \in [0.1,\ 18]$ and the fitted data variables $\hat{\mathbf{z}} \in [0, 55]$, two local minima were observed as shown in Table 11.

Using no initial bounds updating, no updating at each iteration, and a relative convergence tolerance of 0.1%, the problem was solved to global optimality. Table 12 shows the results using different methods of automatically calculating the β values. The Gerschgorin method [23;24] was used to determine the eigenvalues of the interval Hessian matrices. It is interesting to note that no matter how many points with or without interval calculations, that the solution requires about 50 iterations.

The bounds on $\boldsymbol{\theta}$ and the fitted variables needed to allow convergence in the collocation

Obj	θ_1	θ_2	p_1	p_2	Frequency
22.1814	12.29505	8.184225	4.5743×10^{-6}	2.7902×10^{-4}	81.4 %
264.649	6.64074	0.101005	1.3060×10^{-3}	9.0393×10^{-1}	18.6 %

Table 11: Local solutions for Bellman's problem using the integration approach. Frequency refers to the percentage of starting points resulting in the given solution (determined using 1000 random starting points).

β Method	No. of points used	Iterations	CPU sec.
Sampled	15	48	36.16
Sampled	30	49	52.24
Sample/Interval	15	51	40.04
Sample/Interval	30	51	58.91

Table 12: Global optimization solution times for Bellman's problem.

formulation were used in the integration formulation. Calculated β values were used with 30 sampled points, and interval calculations. The global solution (with the same 0.1% convergence) was obtained in 4 iterations and 19.2 CPU sec. Since the model is nonlinear in the states, the integration approach outperforms the collocation. This is not only because the parameters are all that need to be branched on in the integration approach, but also the calculated β values were of order one (with the tight and the expanded initial bounds). Therefore, the gap between the underestimator and the original problem is very small even at the root node in the branch and bound tree.

7.5 Methanol-to-Hydrocarbons Process

This model represents the conversion of methanol to various hydrocarbons. The simplified kinetic model, which appears in Maria[13], is defined as:

$$\begin{aligned} A &\xrightarrow{k_1} B \\ A + B &\xrightarrow{k_2} C \\ C + B &\xrightarrow{k_3} P \\ A &\xrightarrow{k_4} C \\ A &\xrightarrow{k_5} P \\ A + B &\xrightarrow{k_6} P \end{aligned}$$

where A represents the oxygenates, $B = \ddot{C}H_2$, C is the olefins, and P denotes the paraffins, aromatic, and other products. The reaction model is formulated under the assumption of simple kinetics and a quasi-steady state for the intermediate B. The resulting differential equation system is nonlinear in both the states and the parameters, and takes the form:

$$\begin{aligned}\frac{dz_1}{dt} &= -\left(2\theta_1 - \frac{\theta_1 z_2}{(\theta_2+\theta_5)z_1 + z_2} + \theta_3 + \theta_4\right) z_1 \\ \frac{dz_2}{dt} &= \frac{\theta_1 z_1(\theta_2 z_1 - z_2)}{(\theta_2+\theta_5)z_1 + z_2} + \theta_3 z_1 \\ \frac{dz_3}{dt} &= \frac{\theta_1 z_1(z_2 + \theta_5 z_1)}{(\theta_2+\theta_5)z_1 + z_2} + \theta_4 z_1 \\ \mathbf{z}_0 &= [1,\ 0,\ 0] \qquad t \in [0,\ 1.121] \\ 0.1 &\leq \theta_2 + \theta_5\end{aligned} \tag{54}$$

where the state vector, $\mathbf{z}$, is defined as $[A, C, P]$ and the parameter vector, $\boldsymbol{\theta}$ is defined to be $[k_1, \frac{k_2}{k_3}, k_4, k_5, \frac{k_6}{k_3}]$. The data used in this study was obtained from Maria and Muntean[49]. The last algebraic constraint on the parameter values is necessary since when both θ_2 and θ_5 equal zero, the time derivatives are undefined for the given initial condition.

Collocation Approach

The state profiles were approximated using 4^{th} order polynomials on 3 finite elements with element boundaries at t = [0, 0.2, 0.4]. Using bounds on the parameters of $\boldsymbol{\theta} \in [0, 20]$ and the fitted data variables $\hat{\mathbf{z}}_\mu \in [0, 1]$, very different local solutions were identified (see Table 13). It should be noted that all the local solutions not only show different parameter values, but also predict different reaction schemes than the global solution.

Obj Value	θ_1	θ_2	θ_3	θ_4	θ_5	Frequency
0.10652	5.1981	1.2112	0	0	0	76.3 %
0.60110	12.0889	20	0	0	14.1749	2.3 %
0.62634	15.7927	1.1806	0	0	0.1403	0.7 %
2.91739	11.6113	0	0	0	1.1611	10.9 %
10.22539	1.6111	0	0	20	0.1611	0.2 %

Table 13: Local solutions for the methanol to hydrocarbons process using the collocation approach. The frequency was determined using 1000 random starting points.

The model given by (54) appears to be very nonlinear in both the states and the parameters, but the problem can be reformulated into a simpler set of equations. The substitution,

$$b = \frac{\theta_1\, z_1}{(\theta_2 + \theta_5)\, z_1 + z_2} \tag{55}$$

is introduced. This results in the following differential-algebraic system of equations,

$$
\begin{aligned}
\frac{dz_1}{dt} &= -\theta_1 z_1 - \theta_3 z_3 - \theta_4 z_1 - \theta_2 z_1 b - \theta_5 z_1 b \\
\frac{dz_2}{dt} &= \theta_3 z_1 + \theta_2 z_1 b - z_2 b \\
\frac{dz_3}{dt} &= \theta_4 z_1 + \theta_5 z_1 b + z_2 b \\
0 &= \theta_1 z_1 - \theta_2 z_1 b - \theta_5 z_1 b - z_2 b \\
0.1 &\leq \theta_2 \ + \theta_5 \qquad .
\end{aligned} \tag{56}
$$

All the nonconvex terms in (56) take the form of bilinear and trilinear terms. These are simply relaxed using the sets of linear cuts given by (23) and (24). A global solution was attempted using initial bounds updating on all the variables ($\boldsymbol{\theta}$, $\boldsymbol{\xi}$, and $\mathbf{b}$), branching on the same set of variables, and updating 10 randomly selected variables at each iteration from the set [$\boldsymbol{\xi}$, $\mathbf{b}$]. Table 14 shows results using different levels of initial variable bounds (note that the fitted variables are always absolutely bounded between 0 and 1). The results show that even for tight initial bounds, convergence to the global solution is not achieved in a reasonable time. In each case though, the upper bound at iteration 1000 was the best local solution given in Table 13.

Initial Variable Bounds		Iterations	CPU sec.	Rel. conv
$\hat{\mathbf{z}}_\mu \in [0,\ 1]$	$\mathbf{b} \in [0,\ 10]$	1000	19,125	93 %
$\hat{\mathbf{z}}_\mu \in [\bar{\mathbf{z}}_\mu \pm 0.2]$	$\mathbf{b} \in [0,\ 10]$	1000	17,920	54 %
$\hat{\mathbf{z}}_\mu \in [\bar{\mathbf{z}}_\mu \pm 0.15]$	$\mathbf{b} \in [0,\ 5]$	1000	13,624	43 %

Table 14: Global optimization attempts for the methanol to hydrocarbons process using the collocation approach.

Integration Approach

Using the same bounds on the parameters and the fitted data variables as in the collocation approach, and the reformulation given by (56), two different local solutions were found (see Table 15). Each of the local minima predict a different reaction scheme than the global solution. The second local solution (obj = 2.05476) is not unique in θ_2 and θ_5, and also is not physically possible. Reaction 1 (θ_1) needs to occur since this is the only source of component B. This solution is made possible through the pseudo-steady state approximation which results in the initial concentration of B being allowed to vary and take on a non-zero value.

Using initial variable bounds updating and branching on the parameters, with the bounds on the variables specified above, the problem was solved to global optimality. The values of β were calculated using the sampling method with interval analysis, and the Gerschgorin method[23;24] was used to determine the minimum eigenvalues of the interval Hessian matrices. Table 16 provides the results using different numbers of sampled points. The average and deviation of three runs is given. There is a significant difference between runs using the same number of sampled points, but this decreases with larger sample sizes. As the number

Obj Value	θ_1	θ_2	θ_3	θ_4	θ_5	Frequency
0.10693	5.2406	1.2176	0	0	0	91.3 %
2.05476	0	8.6927	1.8477	1.6713	17.6135	8.7 %

Table 15: Local solutions for methanol to hydrocarbons process using the integration approach. The solution frequency was determined using 1000 random starting points.

of points doubles, the variation is reduced by almost a factor of three. Since this problem contains five parameters, the use of additional sampled points is needed to give consistent results. In every case, the global solution was obtained even using a very small sample size (10 points, or only 2 per dimension).

# sampled points	Iterations	CPU sec.
50/10	606 ± 151	1798.0 ± 436.2
100/20	639 ± 65	2610.1 ± 264.7

Table 16: Global optimization solution times for the methanol to hydrocarbons process using the integration approach. The number of sampled points refers to the number used initially and the minimum number used in each subsequent region.

For this problem which is nonlinear in the states, the integration approach outperforms the collocation approach by converging to the global solution with reasonable computation effort. Even though the nonlinearities in the states take the form of bilinear terms (i.e., $z_1\, z_2$), this suffices for the collocation approach to fail to achieve convergence. Another interesting observation is that the collocation approach has introduced new local solutions which do not appear using the integration approach.

7.6 Lotka-Voltera Problem

This problem has been studied by Luus[12]. This model is a representation of the predator-prey model used in ecology. The system is described by two differential equations:

$$\begin{aligned} \frac{dz_1}{dt} &= \theta_1\, z_1\, (1 - z_2) \qquad (57) \\ \frac{dz_2}{dt} &= \theta_2\, z_2\, (z_1 - 1) \\ \mathbf{z}_0 &= [1.2,\, 1.1] \qquad t \in [0,\, 10] \end{aligned}$$

where z_1 represents the population of the prey, and z_2 the population of the predator. The solutions to these equations are cyclic in nature and out of phase with each other. The data used in the study was generated using values for the parameters $\boldsymbol{\theta} = [3,\, 1]$ with a small amount of normally distributed random error with $\sigma = 0.01$ and zero mean added to the observations.

Collocation Approach

The state profiles were approximated by 5th order polynomials on 5 equally spaced finite elements. Using bounds on the parameters of $\boldsymbol{\theta} \in [0.1, 10]$ and $\mathbf{z}_\mu$, $\boldsymbol{\xi} \in [0.5, 1.5]$ a number of local solutions were obtained as shown in Table 17. The most prevalent local solution found (nearly half of the time) was not the global solution, but one of the worst suboptimal solutions.

Objective	θ_1	θ_2	Frequency
1.3194×10^{-3}	3.2521	0.9183	7.4 %
4.6369×10^{-1}	10	2.9247	8.4 %
6.2618×10^{-1}	8.2756	7.1832	3.5 %
6.4865×10^{-1}	6.9116	2.1888	7.0 %
6.6071×10^{-1}	1.7534	5.5731	5.6 %
7.0987×10^{-1}	3.5204	1.8164	4.9 %
7.8925×10^{-1}	1.3122	0.6552	4.1 %
8.2521×10^{-1}	0.7438	0.2636	4.8 %
8.4591×10^{-1}	0.3023	0.1	49.9 %
9.3085×10^{-1}	6.4652	3.0438	2.4 %

Table 17: Local solutions for the Lotka-Voltera problem using the collocation approach. The frequency was determined using 1000 random starting points.

In order to underestimate the model given by (57) it is necessary to rewrite it in the following form:

$$\begin{aligned} \frac{dz_1}{dt} &= \theta_1 \, z_1 \; - \; \theta_1 \, z_1 \, z_2 \\ \frac{dz_2}{dt} &= \theta_2 \, z_2 \, z_1 \; - \; \theta_2 \, z_2 \end{aligned} \tag{58}$$

The nonconvexities in the above equation take the form of either bilinear or trilinear terms. These are relaxed using a series of linear cuts that, in the case of the bilinear terms, represent the convex hull. Even though the underestimation is the tightest possible, convergence to the global solution was not obtained in a reasonable amount of time. The best run involved the use of tighter bounds on the fitted data variables of $\hat{z}_\mu \in [\bar{z}_\mu \pm 0.1]$, updating all variables initially, and 20 randomly chosen $\boldsymbol{\xi}$ variables at each iteration, and branching on the $\boldsymbol{\theta}$ variables. After 1000 iterations the lower bound on the global solution was 2.36886×10^{-4}. The current upper bound was the best known local solution given in Table 17. This run took over 100,000 CPU sec.

Integration Approach

Using the same bounds as for the collocation approach, more than 20 different local solutions were determined. Table 18 shows those solutions that were found more than 2% of the time. The most prevalent local solution found is not the global solution, but the third best solution. As an illustration of just how many local solutions exist, Figure 5 shows slices through the objective function surface at the parameter values of the two best solutions.

Objective	θ_1	θ_2	Frequency
1.2493×10^{-3}	3.2434	0.9209	11.1 %
1.9201×10^{-2}	10	6.4962	6.0 %
5.1010×10^{-1}	8.7871	2.1191	14.3 %
7.0951×10^{-1}	6.2412	2.1148	7.8 %
7.1156×10^{-1}	9.7073	1.0804	5.1 %
7.3607×10^{-1}	3.4852	1.8187	8.3 %
7.4925×10^{-1}	10	7.7757	2.8 %
7.8923×10^{-1}	1.3123	0.6551	3.8 %
7.8941×10^{-1}	10	5.2323	5.0 %
8.2520×10^{-1}	0.7438	0.2636	2.5 %
8.3022×10^{-1}	10	4.5547	5.2 %
8.4384×10^{-1}	1.9292	5.0161	4.3 %
8.9296×10^{-1}	7.0100	4.0108	9.8 %
9.1397×10^{-1}	7.5827	4.7761	4.4 %
9.2118×10^{-1}	7.8125	4.1773	3.5 %

Table 18: Local Solutions for the Lotka-Voltera problem using the integration approach. The frequency was determined using 1000 random starting points.

The problem was solved to global optimality in a reasonable amount of time. No initial bounds updating was performed and an absolute tolerance of 1×10^{-5} was used for every run. Table 19 shows results with different constant values of β. Smaller values converged to the global solution quicker, but a value of 0.1 caused the algorithm to fail. The results presented, for the smaller values, are the average of three different runs. Since the values of β used may not be totally valid, the lower bounding problem may still have multiple minima. As a result, the number of iterations required may vary depending on whether the global minimum of the lower bounding function is identified in each region. The values determined by sampling the space ranged in magnitude from 1×10^{-1} to 1×10^{8}.

The problem was also solved using the β updating methods. These results, the average and standard deviation of three runs, are presented in Table 20. It is apparent that updating the β values has a pronounced effect of the convergence rate for this problem. As the number of points used in the calculation is increased, there is a slight difference in the required number of iterations. This shows that even with a small sample size, it is possible to generate reasonably valid β values. Also, the larger the number of sampled points is, the less variation between runs exists.

This is another example in which the states appear nonlinearly and it cannot be solved to global optimality using the collocation approach. The combination of the number of variables required to achieve a reasonable approximation, with the number of local minima

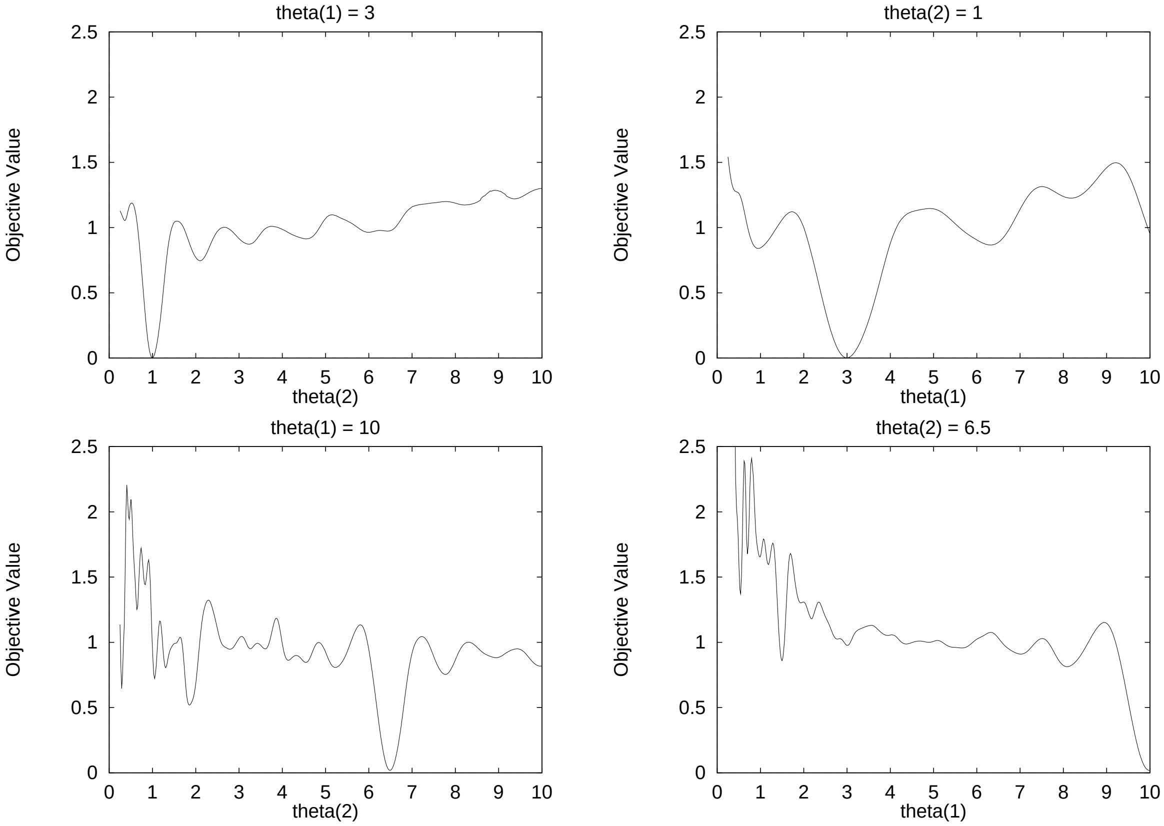


Figure 5: Cuts through the objective surface for the Lotka-Voltera Problem.

β	Iters	CPU sec.
Sampled	9007	9689.67
10.0	2515	6225.69
5.0	1312	2856.45
1.0	336	769.07
0.5	161	367.26
0.1	failed	-

Table 19: Results for the Lotka-Voltera Problem using the integration approach with constant β values. Sampled values were determined using 1000 integration points.

	Sampled		Sampled/Interval	
No. of points	Inter.	CPU sec.	Inter.	CPU sec.
25/10	123 ± 14	331.04 ± 56.54	168 ± 4	396.25 ± 20.00
50/25	145 ± 6	540.04 ± 41.50	195 ± 3	699.73 ± 13.24
100/50	152 ± 3	884.14 ± 49.50	199 ± 2	1119.38 ± 10.16

Table 20: Results for the Lotka-Voltera problem using the integration approach with updated β values.

present were the primary reasons for its failure. On the other hand, the integration approach with automatic β calculation performed flawlessly. In every case, even with a small sample size and despite the huge number of local minima, the approach was able to converge to the global minima in a very reasonable time.

8 Comparisons between Methods

From the experience gained through the solution of the presented example problems, the following comparisons between the methods can be made:

- Since the formulation used in the collocation approach is strictly algebraic in nature, each local solution requires much less time than the integration approach. The integration method requires that a dynamic system be integrated with sensitivity analysis for each constraint evaluation required by the local solver. Also, the calculation of the β values requires integrations with second order sensitivity evaluations. Therefore, the integration method must be able to converge in fewer iterations to achieve the same computational effort.

- The choice of polynomial orders and the number and size of the elements can be rather arbitrary in the collocation approach. The choice of these parameters directly influences the error involved in the approximation of the state profiles. The greater the polynomial order and the number of elements there are, the better the approximation becomes, but the number of nonlinear and nonconvex variables increases. In the parameter estimation problem presented in this paper, the observations can be used to make informed choices for these parameters, but in the general case this is not possible. Methods exist that attempt to control the approximation error[50], but these approaches further complicate an already large formulation. The integration approach does not suffer from these drawbacks since error control is handled directly in the integration routine.

- How the states enter into the dynamic system has a direct impact on the effort required by the collocation approach. For systems which are linear in the states the collocation approach performs well, even better than the integration method. For problems which are nonlinear in the states, the collocation approach performs poorly, even failing to achieve convergence. In the linear case, the bounds on the collocation variables, $\boldsymbol{\xi}$, appear in the underestimating formulation, but do not have a separate or distinct effect. The bounds on the parameters control the quality of the underestimation, and thus branching on these variables is only required. In the nonlinear case, the bounds on the $\boldsymbol{\xi}$ variables has a distinct effect and thus need to be included in the branching set. This greatly increases the size of the set and adversely affects the convergence rate. The integration approach does not depend on how the state variables appear in the system. In all cases only the bounds on the parameters affect the quality of the underestimator, and only these variables need to be branched on. As a result, the integration approach achieved convergence for all the problems tested.

- The initial bounds selected for the fitted data variables has a dramatic effect on the collocation approach. The bounds on these variables, which do not appear in any nonconvex terms, determine the bounds on the collocation coefficients though the bounds updating problems. There is a direct relationship, that is, the tighter the initial bounds on the fitted variables are, the tighter the calculated bounds on the collocation coefficients become. For one problem, even though it is linear in the states, doubling the initial range of the fitted variables, resulted in a three fold increase in the computational expense required. The integration approach is not affected by the choice of initial bounds on the fitted variables. These bounds do not appear, either directly or indirectly, in the formulation of the underestimator.

- The underestimation schemes, as implemented, for the collocation approach, are convex and therefore the approach is guaranteed to converge to the global solution of the algebraically converted problem. The underestimation scheme for the integration approach is theoretically convex if the values used for the β parameters are large enough, as shown by (38). We can theoretically prove the validity of the β values for cases in which the elements of the Hessian matrices can be determined analytically. This is not possible for all problems. Even so, the sampling methods with interval analysis offer reasonably accurate approximations. Using these approaches the algorithm has always converged to the global solution.

9 Conclusions

In this paper, two different global optimization approaches were developed for the estimation of parameters in differential-algebraic models from time series data. Both approaches are based on a branch and bound framework. The first uses orthogonal collocation on finite elements to convert the dynamical system into a set of algebraic equations. The formulation is then solved to global optimality using a modification of the αBB[23;24] for general twice-differentiable NLPs. The second approach uses an integration routine for the calculation of the state profiles. A theoretically convex underestimator was developed for this type of formulation, as well as methods of practical implementation. Various example problems were solved to illustrate the theoretical and computational aspects of both approaches.

Acknowledgment

The authors gratefully acknowledge financial support from the National Science Foundation.

References

(1) Bard, Y. *Nonlinear Parameter Estimation*; Academic Press: New York, 1974.

(2) Bard, Y.; Lapidus, L. Kinetics Analysis by Digital Parameter Estimation. *Catalysis Reviews* **1968**, *2* (1), 67.

(3) Stewart, W. E.; Caracotsios, M.; Sorensen, J. P. Parameter Estimation from Multiresponse Data. *AIChE J.* **1992**, *38* (5), 641.

(4) Bellman, R.; Jacquez, J.; Kalaba, R.; Schwimmer, S. Quasilinearization and the Estimation of Chemical Rate Constraints from Raw Kinetic Data. *Mathematical Biosciences* **1967**, *1*, 71.

(5) Hwang, M.; Seinfeld, J. H. A new algorithm for the estimation of parameters in ordinary differential equations. *AIChE J.* **1972**, *18* (1), 90.

(6) Bilardello, P.; Joulia, X.; LeLann, J. M.; Delmas, H.; Koehret, B. A general strategy for parameter estimation in differential-algebraic systems. *Comp. Chem. Engng.* **1993**, *17* (5/6), 517.

(7) Kim, I. W.; Liebman, M. J.; Edgar, T. F. A sequential error-in-variables method for nonlinear dynamic systems. *Comp. Chem. Engng.* **1991**, *15* (9), 663.

(8) Kalogerakis, N.; Luus, R. Simplification of quasilinearization method for parameter estimation. *AIChE J.* **1983**, *29* (5), 858.

(9) Wang, B.; Luus, R. Increasing the size of region of convergence for parameter estimation through the use of shorter data–length. *Int. J. Control* **1980**, *31* (5), 947.

(10) Luus, R.; Jaakola, T. H. I. Optimization by direct search and systemic reduction of the size of search region. *AIChE J.* **1973**, *19* (4), 760.

(11) Park, T.; Froment, G. F. A Hybrid Genetic Algorithm for the Estimation of Parameters in detailed kinetic models. *Comp. Chem. Engng.* **1998**, *22*, S103.

(12) Luus, R. Parameter Estimation of Lotka-Volterra Problem by Direct Search Optimization. *Hung. J. Ind. Chem.* **1998**, *26*, 287.

(13) Maria, G. An Adaptive Strategy for Solving Kinetic Model Concomitant Estimation-Reduction Problems. *Can. J. Chem. Eng.* **1989**, *67*, 825.

(14) Wolf, D.; Moros, R. Estimating rate constants of heterogeneous catalytic reactions without supposition of rate determining surface steps – an application of a genetic algorithm. *Comp. Chem. Engng.* **1997**, *52* (7), 1189.

(15) Villadsen, J.; Michelsen, M. L. *Solution of Differential Equation Models by Polynomial Approximation*; Prentice-Hall, Inc., 1978.

(16) Van Den Brosch, B.; Hellinckx, L. A New Method for the estimation of parameters in differential equations. *AIChE J.* **1974**, *20* (2), 250.

(17) Baden, N.; Villadsen, J. A family of collocation based methods for parameter estimation in differential equations. *The Chemical Engineering Journal* **1982**, *23*, 1.

(18) Tjoa, T. B.; Biegler, L. T. Simultaneous Solution and Optimization Strategies for Parameter Estimation of Differential-Algebraic Equation Systems. *I&EC Res.* **1991**, *30*, 376.

(19) Liebman, M. J.; Edgar, T. F.; Lasdon, L. S. Efficient Data Reconciliation and Estimation for Dynamic Processes using Nonlinear Programming Techniques. *Comp. Chem. Engng.* **1992**, *16* (10/11), 963.

(20) Androulakis, I.; Maranas, C. D.; Floudas, C. A. αBB : A Global Optimization Method for General Constrained Nonconvex Problems. *Journal of Global Optimization* **1995**, *7*, 337.

(21) Adjiman, C. S.; Androulakis, I. P.; Maranas, C. D.; Floudas, C. A. A Global Optimization Method, αBB, for Process Design. *Computers chem. Engng, Suppl.* **1996**, *20*, S419.

(22) Adjiman, C. S.; Floudas, C. A. Rigorous Convex Underestimators for General Twice–Differentiable Problems. *Journal of Global Optimization* **1996**, *9*, 23.

(23) Adjiman, C. S.; Androulakis, I. P.; Floudas, C. A.; Neumaier, A. A Global Optimization Method, αBB, for General Twice–Differentiable NLPs – I. Theoretical Advances. *Comp. Chem. Engng.* **1998**, *22* (9), 1137.

(24) Adjiman, C. S.; Androulakis, I. P.; Floudas, C. A. A Global Optimization Method, αBB, for General Twice–Differentiable NLPs – II. Implementation and Computational Results. *Comp. Chem. Engng.* **1998**, *22* (9), 1159.

(25) Esposito, W. R.; Floudas, C. A. Global Optimization in Parameter Estimation of Nonlinear Algebraic Models via the Error-in-Variables Approach. *I&EC Res.* **1998**, *37* (5), 1841.

(26) Finlayson, B. A. *The Method of Weighted Residuals and Variational Principles*; Academic Press, New York, 1972.

(27) Cuthrell, J. E.; Biegler, L. T. On the optimization of differential-algebraic process systems. *AIChE J.* **1987**, *33* (8), 1257.

(28) Tieu, D.; Cluett, W. R.; Penlidis, A. A comparison of collocation methods for solving dynamic optimization problems. *Comp. Chem. Engng.* **1995**, *19* (4), 375.

(29) Floudas, C. A. *Deterministic Global Optimization: Theory, Methods and Applications*; Nonconvex Optimization and its Applications Kluwer Academic Publishers, 2000.

(30) Al-Khayyal, F. A.; Falk, J. E. Jointly Constrained Biconvex Programming. *Maths Ops Res.* **1983**, *8*, 273.

(31) McCormick, G. P. Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimations Problems. *Mathematical Programming* **1976**, *10*, 147.

(32) Maranas, C.; Floudas, C. Finding All Solutions of Nonlinearly Constrained Systems of Equations. *Journal of Global Optimization* **1995**, *7* (2), 143.

(33) Quesada, I.; Grossmann, I. A Global Optimization Algorithm for Linear Fractional and Bilinear Programs. *Journal of Global Optimization* **1995**, *6*, 39.

(34) Zamora, J. M.; Grossmann, I. E. A Branch and Contract Algorithm for Problems with Concave Univariate, Bilinear and Linear Fractional Terms. *Journal of Global Optimization* **1999**, *14*, 217.

(35) Maranas, C. D.; Floudas, C. Global Minimum Potential Energy Conformations of Small Molecules. *Journal of Global Optimization* **1994**, *4*, 135.

(36) Pontryagin, L. S. *Ordinary Differential Equations*; Addison-Wesley Publishing Co., 1962 Translated from Russian by L. Kacinskas and W. B. Counts.

(37) Brenan, K. E.; L., C. S.; Petzold, L. R. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*; Classics in Applied Mathematics Society for Industrial and Applied Mathematics: Philadelphia, 1996.

(38) Caracotsios, M.; Stewart, W. E. Sensitivity Analysis of Initial Value Problems with Mixed ODEs and Algebraic Equations. *Comp. Chem. Engng.* **1985**, *9* (4), 359.

(39) Vassiliadis, V. S.; Sargent, R. W. H.; Pantelides, C. C. Solution of a class of multistage dynamic Optimization Problems 1. Problems without Path Constraints. *I&EC Res.* **1994**, *33*, 2111.

(40) Schweiger, C. A.; Floudas, C. A. *MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer, Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization*; Princeton University 3rd ed. 1998.

(41) Vassiliadis, V. S.; Balsa-Canto, E.; Banga, J. R. Second Order Sensitivities of Systems of Differential-Algebraic Equations with Application to Optimal Control Problems. AIChE National Meeting, Miami, November 1998.

(42) Gill, P. E.; Murray, W.; Saunders, M. A. *User's Guide for SNOPT 5.3: A Fortran Package for Large-Scale Nonlinear Programming*; Systems Optimization Laboratory Department of Operations Research, Stanford University 1997.

(43) Murtagh, B. A.; Saunders, M. A. *MINOS 5.4 User's Guide*; Systems Optimization Laboratory Department of Operations Research, Stanford University 1993 Technical Report SOL 83-20R.

(44) Gill, P. E.; Murray, W.; Saunders, M.; Wright, M. H. *User's Guide for NPSOL (Version 4.0)*; Systems Optimization Laboratory Department of Operations Research, Stanford University 1986 Technical Report SOL 86-2.

(45) Jarvis, R. B.; Pantelides, C. C. *DASOLV: A Differential-Algebraic Equation Solver*; Center for Process Systems Engineering Imperial College of Science, Technology, and Medicine, London 1992 Version 1.2.

(46) Floudas, C. A.; Pardalos, P. M.; Adjiman, C. S.; Esposito, W. R.; Gümüş, Z.; Harding, S.; Klepeis, J.; Meyer, C.; Schweiger, C. *Handbook of Test Problems in Local and Global Optimization*; Kluwer Academic Publishers, 1999.

(47) Quesada, I.; Grossmann, I. Global Optimization of Bilinear Process Networks with Multicomponent Flows. *Comp. Chem. Engng.* **1995**, *19* (12), 1219.

(48) Varah, J. M. A spline least squares method for numerical parameter estimation in differential equations. *SIAM J. Sci. Stat. Comput.* **1982**, *3* (1), 28.

(49) Maria, G.; Muntean, O. Model reduction and kinetic parameters identification for the methanol conversion to olefins. *Chem. Eng. Science* **1987**, *42* (6), 1451.

(50) Vasantharajan, S.; Biegler, L. T. Simultaneous Strategies for the Optimization of Differential-Algebraic Systems with Enforcement of Error Criteria. *Comp. Chem. Engng.* **1990**, *14* (10), 1083.

A Data for Example Problems

μ	t_{mu}	$\bar{z}_1$	$\bar{z}_2$
1	0.1	0.606	0.373
2	0.2	0.368	0.564
3	0.3	0.223	0.647
4	0.4	0.135	0.669
5	0.5	0.082	0.656
6	0.6	0.050	0.624
7	0.7	0.030	0.583
8	0.8	0.018	0.539
9	0.9	0.011	0.494
10	1.0	0.007	0.451

Table 21: Data for Example 1.

		Error Free			Error Added		
μ	t_μ	$\bar{z}_1$	$\bar{z}_2$	$\bar{z}_3$	$\bar{z}_1$	$\bar{z}_2$	$\bar{z}_3$
1	0.05	0.8241	0.0937	0.0821	0.8261	0.0917	0.0826
2	0.10	0.6852	0.1345	0.1802	0.6782	0.1335	0.1772
3	0.15	0.5747	0.1654	0.2598	0.5721	0.1644	0.2628
4	0.20	0.4867	0.1899	0.3233	0.4817	0.1939	0.3213
5	0.25	0.4166	0.2094	0.3738	0.4226	0.2111	0.3598
6	0.30	0.3608	0.2249	0.4141	0.3698	0.2229	0.4201
7	0.35	0.3164	0.2373	0.4461	0.3114	0.2313	0.4511
8	0.40	0.2810	0.2472	0.4717	0.2710	0.2398	0.4797
9	0.45	0.2529	0.2550	0.4920	0.2499	0.2510	0.4990
10	0.50	0.2304	0.2613	0.5082	0.2354	0.2703	0.5122
11	0.55	0.2126	0.2662	0.5210	0.2216	0.2602	0.5200
12	0.60	0.1984	0.2702	0.5313	0.1974	0.2732	0.5281
13	0.65	0.1870	0.2733	0.5395	0.1890	0.2733	0.5305
14	0.70	0.1780	0.2759	0.5460	0.1780	0.2769	0.5500
15	0.75	0.1709	0.2779	0.5511	0.1729	0.2709	0.5601
16	0.80	0.1651	0.2794	0.5553	0.1701	0.2754	0.5533
17	0.85	0.1606	0.2807	0.5585	0.1606	0.2797	0.5485
18	0.90	0.1570	0.2817	0.5612	0.1490	0.2817	0.5612
19	0.95	0.1541	0.2825	0.5632	0.1531	0.2825	0.5632
20	1.00	0.1518	0.2832	0.5649	0.1568	0.2792	0.5599

Table 22: Data for Example 2.

μ	t_μ	$\bar{z}_1$	$\bar{z}_2$
1	0.025	0.7307	0.1954
2	0.050	0.5982	0.2808
3	0.075	0.4678	0.3175
4	0.100	0.4267	0.3047
5	0.125	0.3436	0.2991
6	0.150	0.3126	0.2619
7	0.175	0.2808	0.2391
8	0.200	0.2692	0.2210
9	0.225	0.2210	0.1898
10	0.250	0.2122	0.1801
11	0.300	0.1903	0.1503
12	0.350	0.1735	0.1030
13	0.400	0.1615	0.0964
14	0.450	0.1240	0.0581
15	0.500	0.1190	0.0471
16	0.550	0.1109	0.0413
17	0.650	0.0890	0.0367
18	0.750	0.0820	0.0219
19	0.850	0.0745	0.0124
20	0.950	0.0639	0.0089

Table 23: Data for Example 3.

μ	t_μ	$\bar{z}$
1	1.0	1.4
2	2.0	6.3
3	3.0	10.4
4	4.0	14.2
5	5.0	17.6
6	6.0	21.4
7	7.0	23.0
8	9.0	27.0
9	11.0	30.5
10	14.0	34.4
11	19.0	38.8
12	24.0	41.6
13	29.0	43.5
14	39.0	45.3

Table 24: Data for Example 4.

μ	t_μ	$\bar{z}_1$	$\bar{z}_2$	$\bar{z}_3$
1	0.050	0.461	0.114	0.018
2	0.065	0.426	0.135	0.035
3	0.080	0.383	0.157	0.045
4	0.123	0.305	0.194	0.047
5	0.233	0.195	0.231	0.084
6	0.273	0.170	0.234	0.095
7	0.354	0.139	0.228	0.111
8	0.397	0.112	0.228	0.134
9	0.418	0.112	0.226	0.168
10	0.502	0.090	0.220	0.148
11	0.553	0.082	0.214	0.157
12	0.681	0.066	0.178	0.206
13	0.750	0.053	0.188	0.206
14	0.916	0.043	0.183	0.214
15	0.937	0.041	0.184	0.213
16	1.122	0.029	0.166	0.230

Table 25: Data for Example 5.

μ	t_μ	$\bar{z}_1$	$\bar{z}_2$
1	1.0	0.7990	1.0758
2	2.0	0.8731	0.8711
3	3.0	1.2487	0.9393
4	4.0	1.0362	1.1468
5	5.0	0.7483	1.0027
6	6.0	1.0024	0.8577
7	7.0	1.2816	1.0274
8	8.0	0.8944	1.1369
9	9.0	0.7852	0.9325
10	10.0	1.1527	0.9074

Table 26: Data for Example 6.

B α Expressions for Bellman's Problem

The terms and the expressions used for α for the model in its original form are:

$$\begin{aligned}
e^{-\theta_2} z^2 &\longrightarrow -\frac{1}{2}\left\{\frac{1}{2}\,(z^U)^2 + 1 - \frac{1}{2}\sqrt{(z^U)^4 + 12\,(z^U)^2 + 4}\right\} e^{-\theta^L} \\
-e^{-\theta_2} z^2 &\longrightarrow -\frac{1}{2}\left\{-\frac{1}{2}\,(z^U)^2 - 1 - \frac{1}{2}\sqrt{(z^U)^4 + 12\,(z^U)^2 + 4}\right\} e^{-\theta^L} \\
e^{-\theta_1}(c_1 - z)(c_2 - z)^2 &\longrightarrow -\frac{1}{2}\left\{-\frac{1}{2}\,(z^L)^3 + A\,(z^L)^2 - B\,z^L + C\right. \\
&\qquad \left. -\frac{1}{2}\sqrt{(z^L)^6 - D\,(z^L)^5 + E\,(z^L)^4 - F\,(z^L)^3 + G\,(z^L)^2 + H\,z + I}\right\} e^{-\theta^L} \\
-e^{-\theta_1}(c_1 - z)(c_2 - z)^2 &\longrightarrow -\frac{1}{2}\left\{\frac{1}{2}\,(z^L)^3 - A\,(z^L)^2 + B\,z^L - C\right. \\
&\qquad \left. -\frac{1}{2}\sqrt{(z^L)^6 - D\,(z^L)^5 + E\,(z^L)^4 - F\,(z^L)^3 + G\,(z^L)^2 + H\,z + I}\right\} e^{-\theta^L}
\end{aligned}$$

where the constants A through I are defined to be:

$$\begin{aligned}
A &= 2\,c_1 + c_2 \\
B &= c_2^2 + 2\,c_1\,c_2 + 6 \\
C &= c_1\,c_2^2 + 2\,c_1 + 4\,c_2 \\
D &= 4\,c_2 + 2\,c_1 \\
E &= c_1^2 + 8\,c_1\,c_2 + 6\,c_2^2 + 24 \\
F &= 4\,c_1^2\,c_2 + 12\,c_1\,c_2^2 + 4\,c_2^3 + 32\,c_1 + 64\,c_2 \\
G &= 12\,c_1^2 + 6\,c_1^2\,c_2^2 + 8\,c_2^3\,c_1 + 36 + 72\,c_1\,c_2 + c_2^4 + 60\,c_2^2 \\
H &= 2\,c_2^4\,c_1 + 4\,c_1^2\,c_2^3 + 48\,c_1\,c_2^2 + 24\,c_1^2\,c_2 + 24\,c_2^3 + 24\,c_1 + 48\,c_2 \\
I &= 12\,c_1^2\,c_2^2 + 4\,c_1^2 + 16\,c_2^2 + 4\,c_2^4 + 8\,c_1\,c_2^2 + 16\,c_2\,c_1 + c_1^2\,c_2^4
\end{aligned}$$

The terms and the expressions used for α for the expanded model are:

$$\begin{aligned}
e^{-\theta_1} z^3 &\longrightarrow -\frac{1}{2}\left\{\frac{1}{2}(z^U)^2 + 3 - \sqrt{(z^U)^4 + 24\,(z^U)^2 + 36}\right\} z^U e^{-\theta^L} \\
-e^{-\theta_1} z^3 &\longrightarrow -\frac{1}{2}\left\{-\frac{1}{2}(z^U)^2 - 3 - \sqrt{(z^U)^4 + 24\,(z^U)^2 + 36}\right\} z^U e^{-\theta^L} \\
e^{-\theta_2} z^2 &\longrightarrow -\frac{1}{2}\left\{\frac{1}{2}\,(z^U)^2 + 1 - \frac{1}{2}\sqrt{(z^U)^4 + 12\,(z^U)^2 + 4}\right\} e^{-\theta^L} \\
-e^{-\theta_2} z^2 &\longrightarrow -\frac{1}{2}\left\{-\frac{1}{2}\,(z^U)^2 - 1 - \frac{1}{2}\sqrt{(z^U)^4 + 12\,(z^U)^2 + 4}\right\} e^{-\theta^L} \\
e^{-\theta_2} z &\longrightarrow -\frac{1}{2}\left\{\frac{1}{2}\,(z^U) - \frac{1}{2}\sqrt{(z^U)^2 + 4}\right\} e^{-\theta^L} \\
-e^{-\theta_2} z &\longrightarrow -\frac{1}{2}\left\{-\frac{1}{2}\,(z^U) - \frac{1}{2}\sqrt{(z^U)^2 + 4}\right\} e^{-\theta^L}
\end{aligned}$$

C Sample Input File

```
# Lotka-Volterra Predator-Prey model taken from Luus 1998,
# using the integration based approach.
# Example 7.6

Data
nxvar     22
nzvar     2
nfun      1
ndyncon   42
nbterm    0
ncterm    1
nnterm    0
nuterm    0
ninter    0

Name declaration
set   mu(1:10),m(1:2),p(1:2)
xvar  k(p),zhat(mu,m)
zvar  z(m)
fun   f
cterm obj
dyncon dae1,dae2,err1(m)[mu],err2(m)[mu]
param zbar(mu,m) = { 0.7990, 1.0758,\
                    0.8731, 0.8711,\
                    1.2487, 0.9393,\
                    1.0362, 1.1468,\
                    0.7483, 1.0027,\
                    1.0024, 0.8577,\
                    1.2816, 1.0274,\
                    0.8944, 1.1369,\
                    0.7852, 0.9325,\
                    1.1527, 0.9074}

branch k(p)

Options
bcalc standard
epsr 1e-3
epsa 1e-5
epsf 1e-6
epso 1e-6
beta initial points 100
beta points 50
```

```
Solver Options
nlpsolver SNOPT
daesolver DASOLV

Terms
obj = << mu| << m| (zhat(mu,m) - zbar(mu,m))^2 >> >>

Dynamic constraints
timep = {0,1,2,3,4,5,6,7,8,9,10}
initial conditions z(m)

z(1) stp = 1.2
z(2) stp = 1.1

dae1 .. z'(1) - k(1)*z(1)*(1 - z(2)) = 0
dae2 .. z'(2) - k(2)*z(2)*(z(1) - 1) = 0

err1(m)[mu] .. zhat(mu,) - z(m) <= 0
err2(m)[mu] .. -zhat(mu,) + z(m) <= 0

Functions
f .. obj

Bounds
# Parameter Bounds
k(1) lbd = 0
k(1) ubd = 10
k(2) lbd = 0
k(2) ubd = 10

# Fitted variable bounds
zhat(mu,m) lbd = 0.5
zhat(mu,m) ubd = 1.5
```