

# Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems

Victor M. Zavala, Carl D. Laird, Lorenz T. Biegler\*

Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Received 28 February 2007; received in revised form 8 May 2007; accepted 17 May 2007

Available online 25 May 2007

## Abstract

Multi-scenario optimization is a convenient way to formulate and solve multi-set parameter estimation problems that arise from errors-in-variables-measured (EVM) formulations. These large-scale problems lead to nonlinear programs (NLPs) with specialized structure that can be exploited by the NLP solver in order to obtain more efficient solutions. Here we adapt the IPOPT barrier nonlinear programming algorithm to provide efficient parallel solution of multi-scenario problems. The recently developed object oriented framework, IPOPT 3.2, has been specifically designed to allow specialized linear algebra in order to exploit problem specific structure. This study discusses high-level design principles of IPOPT 3.2 and develops a parallel Schur complement decomposition approach for large-scale multi-scenario optimization problems. A large-scale case study example for the identification of an industrial low-density polyethylene (LDPE) reactor model is presented. The effectiveness of the approach is demonstrated through the solution of parameter estimation problems with over 4100 ordinary differential equations, 16,000 algebraic equations and 2100 degrees of freedom in a distributed cluster.

© 2007 Elsevier Ltd. All rights reserved.

**Keywords:** Parameter estimation; Nonlinear programming; Dynamic optimization; Collocation

## 1. Introduction

This study deals with the development of specialized nonlinear programming algorithms for large, structured optimization problems that arise in parameter estimation. The solution of these problems is key to the development of industrially relevant models that go beyond idealized laboratory conditions. In this context, large-scale rigorous models need to be developed that incorporate complex phenomena in order to capture the behavior characterized by actual process data. Moreover, as model development is a repetitive task, efficient methods are needed for fast and accurate estimation of parameters. This task is also essential for data analysis, statistical inference and as a subtask in model discrimination.

Efficient parameter estimation requires structural exploitation of the resulting nonlinear programming problem. These estimation problems present a multi-scenario structure arising from multiple data sets that include both local and global

parameters. While local parameters affect only a particular data set, global parameters affect all sets and, therefore, can be seen as complicating variables between sets. An estimation problem class related to this type of problem is the errors-in-variables-measured (EVM) formulation where the consideration of measurement errors in both input and output variables leads to a significant increase in the degrees of freedom.

While nonlinear multi-scenario optimization formulations can be solved directly with general purpose nonlinear program (NLP) solvers, the problem size can easily become intractable. Traditionally, large-scale structured optimization problems have been handled by specialized problem level decomposition algorithms. In contrast, this study develops an *internal decomposition* for a particular full-space, primal–dual interior-point (IP) algorithm, IPOPT. In this type of algorithm the dominant computational expense is the solution of a large linear system at each iteration. With the proposed decomposition approach, the fundamental interior-point algorithm is not altered, but the linear algebra operations performed by the algorithm are made aware of the problem structure. Therefore, it is possible to develop large-scale decomposition approaches that preserve

\* Corresponding author. Tel.: +1 412 268 2232.

E-mail address: lb01@andrew.cmu.edu (L.T. Biegler).

the desirable convergence properties of the overall NLP algorithm. Similar concepts have also been advanced by Gondzio and Grothey (2004, 2006), primarily for linear, quadratic, and convex programming problems. In this work, we exploit the structure of large, nonconvex multi-scenario problems with a parallel Schur complement decomposition strategy that can be implemented in modern parallel computing architectures.

In the next section we provide a general statement of the parameter estimation problem for both standard least-squares and EVM models. Section 3 then reviews Newton-based barrier methods and discusses their adaptation to multi-scenario problems. Section 4 presents the high-level design of the IPOPT 3.2 software package and describes how the design enables development of internal decomposition approaches without changes to the fundamental algorithm and code. The parallel Schur complement decomposition is implemented within this framework. This approach is demonstrated in Section 5 on large-scale parameter estimation for the identification of an low-density polyethylene (LDPE) reactor model. Results are shown for a parallel implementation of the multi-scenario algorithm. Section 6 then concludes the paper and discusses areas for future work.

## 2. Parameter estimation problem

Consider the differential–algebraic equation (DAE) model

$$\mathbf{F}_k \left[ \frac{dz_k(t)}{dt}, z_k(t), y_k(t), p_k, \Pi \right] = 0,$$

$$\mathbf{G}_k [z_k(t), y_k(t), p_k, \Pi] = 0,$$

$$z_k(0) = z_{0,k}, \quad (1)$$

where  $\mathbf{F}_k(\cdot)$ ,  $\mathbf{G}_k(\cdot)$  are differential and algebraic equations, respectively, defined over a set of operating scenarios or data sets  $k = 1, \dots, \text{NS}$ . Here,  $z_k(t)$  are the differential state variables with initial conditions  $z_{0,k}$ ,  $y_k(t)$  are algebraic state variables and the independent variable  $t$  is either temporal or spatial. In a standard estimation problem, the local and global model parameters  $p_k$  and  $\Pi$ , respectively, are selected to minimize the deviation between the predicted and the measured values of a set of output variables. Here we note that the local parameters may vary in each scenario while the global parameters are common between scenarios. The standard least-squares parameter estimation problem can be stated as

$$\begin{aligned} \min_{\Pi, p_k} & \sum_{k=1}^{\text{NS}} \sum_{i=1}^{\text{NM}_k} (y_k(t_i) - \bar{y}_{k,i})^T \mathbf{V}_y^{-1} (y_k(t_i) - \bar{y}_{k,i}) \\ \text{s.t.} & \\ & \mathbf{F}_k \left[ \frac{dz_k(t)}{dt}, z_k(t), y_k(t), p_k, \Pi \right] = 0, \\ & \mathbf{G}_k [z_k(t), y_k(t), p_k, \Pi] = 0, \\ & \mathbf{H}_k [z_k(t), y_k(t), p_k, \Pi] \leq 0, \\ & z_k(0) = z_{0,k}, \quad k = 1, \dots, \text{NS}, \end{aligned} \quad (2)$$

where  $\mathbf{H}_k(\cdot)$  are general inequality constraints,  $\text{NM}_k$  is the number of measurement locations for a given scenario  $k$ ,  $t_i$  is the

measurement location and  $\bar{y}$  denotes the actual plant measurements. The symbol  $\mathbf{V}_y^{-1}$  denotes a weighting matrix for the algebraic output variables representing an approximation of the inverse covariance matrix. The computational complexity associated to these problems comes from the incorporation of multiple instances of the large-scale DAE model as constraints.

The problem formulation in (2) can be generalized to include a wide variety of objective functions derived from maximum likelihood theory as well as extensions to Bayesian and robust statistics. Because our derivation uses exact second derivatives, no special property is required other than separability of the objective function so that it can be written in the general form:  $\sum_{k=1}^{\text{NS}} \phi(y_k, p_k, \Pi)$ . In particular, the standard least-squares formulation presented in (2) considers errors that are present only in the output variables. It is well known that this approach can produce biased parameters (Moran, 1971; Kendall and Stuart, 1973). On the other hand, the EVM formulation accounts for errors in all the measured variables (both input and output variables) and is particularly useful in finding more reliable global parameters. However, a major difficulty in solving this type of problems is that, since the error is accounted in all the variables, the optimization is performed on both the parameters and the inputs, thus leading to problems with many degrees of freedom. The general EVM formulation resembles standard least-squares (2) except that the inputs in every data set  $k$  become decision variables. Upon addition of terms in the objective function that account for allowed adjustments from measured input variables, the parameter estimation problem becomes

$$\begin{aligned} \min_{\Pi, p_k, u_k} & \sum_{k=1}^{\text{NS}} \sum_{i=1}^{\text{NM}_k} (y_k(t_i) - \bar{y}_{k,i})^T \mathbf{V}_y^{-1} (y_k(t_i) - \bar{y}_{k,i}) \\ & + \sum_{k=1}^{\text{NS}} (u_k - \bar{u}_k)^T \mathbf{V}_u^{-1} (u_k - \bar{u}_k) \\ \text{s.t.} & \\ & \mathbf{F}_k \left[ \frac{dz_k(t)}{dt}, z_k(t), y_k(t), u_k, p_k, \Pi \right] = 0, \\ & \mathbf{G}_k [z_k(t), y_k(t), u_k, p_k, \Pi] = 0, \\ & \mathbf{H}_k [z_k(t), y_k(t), p_k, \Pi] \leq 0, \\ & z_k(0) = z_{0,k}, \quad k = 1, \dots, \text{NS}, \end{aligned} \quad (3)$$

where  $\mathbf{V}_u^{-1}$  is a weighting matrix for the input variables  $u_k$  and  $\bar{u}_k$  are their corresponding measured values; these could include flow rates as well as inlet pressures, temperatures and concentrations. The EVM approach corrects for measurement errors on all these variables and yields less biased parameters. However, this comes at the expense of an increased number of degrees of freedom. Consequently, solutions of EVM problems are often considered to be computationally intensive.

## 3. Solution strategy

Two main approaches have been traditionally used for the solution of the DAE-constrained optimization problems

described in the previous section. First, the sequential or feasible-path approach separates the model solution and optimization tasks. Here, an optimizer updates the parameters and passes them to a DAE solver, which integrates the model equations (Kim et al., 1991). Derivative information required by the optimizer can be obtained through the integration of sensitivity or adjoint equations or by perturbation (Caracotsios and Stewart, 1985). Since this approach requires the repeated solution of the DAE system, it turns out to be computationally inefficient for large-scale models. Moreover, even if the integration task can be parallelized in the presence of multiple data sets (Faber and Wozny, 2003), the optimization task needs to be performed over a reduced but *dense* space of the parameter and input variables, leading to a computational complexity that scales *cubically* with the number of degrees of freedom (Zavala et al., 2007). Nevertheless, because of its relative simplicity in developing solution frameworks from standard optimization and integration algorithms, this approach has been popular for the solution of parameter estimation problems involving DAE models. However, its practical application to large-scale parameter estimation and, particularly EVM problems, has been rather limited.

In the simultaneous or infeasible-path approach, the DAE model solution and optimization tasks are completely coupled by performing a full discretization of the model. With this, the DAE-constrained optimization problem is converted into a large-scale NLP problem with sparse structure. The most important advantage of this approach is that it avoids the continuous and expensive solution of the large-scale model, since the discretized model (algebraic constraints) is solved only once, at the solution of the NLP. The recent potential of this approach has been directly related to the availability of optimization strategies and computational resources able to handle large-scale NLPs (Biegler and Grossmann, 2004). Nowadays, modern nonlinear programming algorithms based on sequential quadratic programming (SQP) and IP methods can efficiently handle large-scale NLPs, thus enabling the solution of challenging DAE-constrained optimization problems (Biegler et al., 2002; Byrd et al., 2000; Benson et al., 2002). Furthermore, application of simultaneous approaches has also become simple and efficient due to the availability of powerful modeling environments (Fourer et al., 1992; Brooke et al., 1998). Moreover, these platforms provide exact first and second derivative information, thus enhancing convergence properties of large-scale NLP algorithms.

Finally, note that the optimization task taking place in the simultaneous approach is performed on a much larger but *sparse* space (degrees of freedom and discretized model variables) that leads to a computational complexity that scales at most *quadratically* with the number of degrees of freedom. That is, the computational complexity of this approach scales better than that of the sequential approach (Zavala et al., 2007). Simultaneous approaches have been demonstrated for the solution of general DAE- and PDAE-constrained optimization problems in many areas of science and engineering, and have been shown to be robust and efficient (Biros and Ghattas, 2003; Betts and Huffman, 2003). On the other hand, these approaches heavily

rely on the efficiency of the optimization solver, require careful initializations and might suffer from numerical difficulties associated to the discretization of highly nonlinear and stiff DAEs. In such cases, feasible-path approaches are expected to be more reliable since they can handle the complexity of the DAE more efficiently.

### 3.1. Model discretization

In this work, a simultaneous approach based on orthogonal collocation on finite elements is used for the solution of the parameter estimation problem. This discretization scheme approximates the differential and algebraic variable profiles by using a family of interpolation polynomials over the entire continuous time domain which is divided into finite elements ( $t_0 < t_1 < \dots < t_{N_{FE}}$ ). Here, we use a monomial basis representation for the differential profiles (Bader and Ascher, 1987) that is particularly attractive since it leads to better condition numbers of the Jacobian matrix

$$z(t) = z_{i-1} + h_i \sum_{q=1}^{NC} \Omega_q \left( \frac{t - t_{i-1}}{h_i} \right) \frac{dz}{dt}_{i,q}, \quad (4)$$

where  $z_{i-1}$  is the value of the differential variable, evaluated at the beginning of element  $i$ ,  $h_i = t_i - t_{i-1}$  is the length of the element  $i$ ,  $dz/dt_{i,q}$  is the value of the first derivative in element  $i$  at collocation point  $q$  and  $\Omega_q$  is an interpolation polynomial of order NC that satisfies,

$$\Omega_q(0) = 0, \quad \Omega'_q(\rho_r) = \delta_{q,r} \quad \text{for } q = 1, \dots, NC,$$

where  $\rho_r$  is the location of the  $r$ th collocation point within each element and  $\delta_{q,r}$  is the Kronecker delta. Continuity of the differential profiles across elements is directly enforced by

$$z_i = z_{i-1} + h_i \sum_{q=1}^{NC} \Omega_q(1) \frac{dz}{dt}_{i,q}. \quad (5)$$

Here, Radau collocation points are used because they stabilize the system more efficiently in the presence of high-index DAEs. The algebraic profiles are approximated using a similar monomial basis representation

$$y(t) = \sum_{q=1}^{NC} \psi_q \left( \frac{t - t_{i-1}}{h_i} \right) y_{i,q}, \quad (6)$$

where  $y_{i,q}$  represents the values of the algebraic variables.  $\psi_q$  is a Lagrange polynomial of order NC satisfying

$$\psi_q(\rho_r) = \rho_{q,r} \quad \text{for } q, r = 1, \dots, NC.$$

Note that the number and length of the finite elements can be adjusted according to the precision required in the approximation. In addition, note that the objective function in problems (2)–(3) may include measurements located at positions that do not coincide with the discretization mesh. In such a case, the

measurement values can be interpolated using the closest point in the mesh.

Upon substitution of the algebraic expressions (4)–(6), problems (2)–(3) can be expressed as large-scale, general NLP problems of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0, \\ & x \geq 0, \end{aligned} \quad (7)$$

where  $x \in \mathfrak{R}^{n_x}$  contains local and global parameters, input variables and all the variables obtained from the discretization of the DAEs corresponding to all the scenarios.

### 3.2. IP methods

In principle, the multi-set or multi-scenario parameter estimation problems (2)–(3) can be solved directly as the general NLP (7), i.e., without exploiting any structure of the problem. However, when the optimization problem becomes too large, specialized approaches able to exploit the multi-scenario problem structure are necessary. In this context, SQP-based strategies have been previously developed that exploit the structure of these problems (Varvarezos et al., 1994; Bhatia and Biegler, 1999). In this study, we present an improved multi-scenario strategy based on a recently developed, primal–dual barrier NLP method called IPOPT.

IPOPT applies a Newton strategy to the optimality conditions that result from the primal–dual barrier subproblem

$$\begin{aligned} \min \quad & f(x) - \mu \sum_{j=1}^{n_x} \ln(x^{(j)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned} \quad (8)$$

for decreasing values of the barrier parameter  $\mu$  and global convergence is promoted by a filter-based line-search strategy (Wächter and Biegler, 2006). Under mild assumptions, the algorithm has global and superlinear convergence properties. Originally developed in FORTRAN, the IPOPT algorithm was recently redesigned to allow for structure dependent specialization of the fundamental linear algebra operations. This new package is implemented in C++ and is freely available through the COIN-OR foundation from the following website: <http://projects.coin-or.org/Ipoppt>. The key step in the IPOPT algorithm is the solution of linear systems derived from the linearization of the first-order optimality conditions (in primal–dual form) of the barrier subproblem. Here, we derive the structured form of these linear systems for the multi-scenario optimization problem and present a specialized decomposition for their solution.

To simplify the resulting decomposition of problems (2)–(3), we introduce, for the global variables  $\Pi$ , additional local linking variables and define them through linear linking constraints for each scenario. Noting how these linking variables and constraints partition among the different scenarios (i.e., data sets) we can rewrite (7) as a generalized multi-scenario problem

of the form

$$\begin{aligned} \min_{x_k, d} \quad & \sum_{k=1}^{NS} f_k(x_k) \\ \text{s.t.} \quad & \left. \begin{aligned} c_k(x_k) &= 0, \\ S_k x_k &\geq 0, \\ D_k x_k - \bar{D}_k d &= 0, \end{aligned} \right\} k = 1, \dots, NS, \end{aligned} \quad (9)$$

where  $x_k$  contains all the parameters and variables corresponding to the discretization of the DAEs for a particular scenario  $k$ ,  $d$  is the vector of linking variables, matrix  $D_k$  extracts the components corresponding to the global parameters  $\Pi$  from the  $x_k$  vector and matrix  $\bar{D}_k$  assigns the extracted components ( $D_k x_k$ ) to the linking variable vector  $d$ . Matrix  $S_k$  extracts the components of  $x_k$  that have bound constraints; if all of the scenarios have the same structure, we can set  $S_1 = \dots = S_{NS}$ .

We note that problem (9) can apply to more general multi-scenario problems than just (2)–(3). For instance, it is possible to incorporate scenarios with heterogeneous structures (e.g. different DAE models). Finally, we note that the above formulation is presented only for implementation purposes, as it now allows us to specify the multi-scenario problem through individual NLP instances (i.e., with the linking constraints in (9) removed).

Using a barrier formulation, problem (9) can be converted to

$$\begin{aligned} \min_{x_k, d} \quad & \sum_{k=1}^{NS} \left\{ f_k(x_k) - \mu \sum_j \ln[(S_k x_k)^{(j)}] \right\} \\ \text{s.t.} \quad & \left. \begin{aligned} c_k(x_k) &= 0, \\ D_k x_k - \bar{D}_k d &= 0, \end{aligned} \right\} k = 1, \dots, NS, \end{aligned} \quad (10)$$

where indices  $j$  correspond to scalar elements of the vector  $(S_k x_k)$ . Defining the Lagrange function of the barrier problem (10)

$$\begin{aligned} \mathcal{L}(x, \lambda, \sigma, d) &= \sum_{k=1}^{NS} \bar{\mathcal{L}}_k(x_k, \lambda_k, \sigma_k, d) \\ &= \sum_{k=1}^{NS} \left\{ \mathcal{L}_k(x_k, \lambda_k, \sigma_k, d) - \mu \sum_j \ln[(S_k x_k)^{(j)}] \right\} \\ &= \sum_{k=1}^{NS} \left\{ f_k(x_k) - \mu \sum_j \ln[(S_k x_k)^{(j)}] \right. \\ &\quad \left. + c_k(x_k)^T \lambda_k + [D_k x_k - \bar{D}_k d]^T \sigma_k \right\} \end{aligned} \quad (11)$$

with multipliers  $\lambda_k$  and  $\sigma_k$ , and  $G_k = \text{diag}(S_k x_k)$  leads to the primal–dual form of the first-order optimality conditions for this equality constrained problem, written as

$$\left. \begin{aligned} \nabla_{x_k} f_k(x_k) + \nabla_{x_k} c_k(x_k) \lambda_k \\ + D_k^T \sigma_k - S_k^T v_k &= 0, \\ c_k(x_k) &= 0, \\ D_k x_k - \bar{D}_k d &= 0, \\ G_k v_k - \mu e &= 0, \\ - \sum_{k=1}^{NS} \bar{D}_k^T \sigma_k &= 0, \end{aligned} \right\} k = 1, \dots, NS, \quad (12)$$

where we define  $e^T = [1, 1, \dots, 1]$ . Writing the Newton step for (12) at iteration  $\ell$  leads to

$$\left. \begin{aligned} \nabla_{x_k x_k} \mathcal{L}_k^\ell \Delta x_k + \nabla_{x_k} c_k^\ell \Delta \lambda_k \\ + D_k^T \Delta \sigma_k - S_k^T \Delta v_k \\ = -(\nabla_{x_k} \mathcal{L}_k^\ell - S_k^T v_k^\ell), \\ \nabla_{x_k} c_k^\ell \Delta x_k = -c_k^\ell, \\ D_k \Delta x_k - \bar{D}_k \Delta d = -D_k x_k^\ell + \bar{D}_k d^\ell, \\ V_k^\ell S_k \Delta x_k + G_k \Delta v_k = \mu e - G_k v_k^\ell, \\ - \sum_{k=1}^{NS} \bar{D}_k^T \Delta \sigma_k = \sum_{k=1}^{NS} \bar{D}_k^T \sigma_k^\ell, \end{aligned} \right\} k = 1, \dots, NS, \quad (13)$$

where the superscript  $\ell$  indicates that the quantity is evaluated at the point  $(x_k^\ell, \lambda_k^\ell, \sigma_k^\ell, v_k^\ell, d^\ell)$ . Eliminating  $\Delta v_k$  from the resulting linear equation gives the primal–dual *augmented system*

$$\left. \begin{aligned} H_k^\ell \Delta x_k + \nabla_{x_k} c_k^\ell \Delta \lambda_k + D_k^T \Delta \sigma_k = -\nabla_{x_k} \bar{\mathcal{L}}_k^\ell, \\ \nabla_{x_k} c_k^\ell \Delta x_k = -c_k^\ell, \\ D_k \Delta x_k - \bar{D}_k \Delta d = -D_k x_k^\ell + \bar{D}_k d^\ell, \\ - \sum_{k=1}^{NS} \bar{D}_k^T \Delta \sigma_k = \sum_{k=1}^{NS} \bar{D}_k^T \sigma_k^\ell, \end{aligned} \right\} k = 1, \dots, NS, \quad (14)$$

where  $H_k^\ell = \nabla_{x_k x_k} \mathcal{L}_k^\ell + S_k^T (G_k^\ell)^{-1} V_k^\ell S_k$ , and  $V_k = \text{diag}(v_k)$ .

According to the IPOPT algorithm (Wächter and Biegler, 2006), the linear system (14) is regularized if necessary by adding diagonal terms. Diagonal elements are added to the block Hessian terms in the *augmented system* to handle nonpositive curvature ( $\delta_1 I$ ) and to the lower right corner in each block to handle temporary dependencies in the constraints ( $-\delta_2 I$ ). Applying these modifications, linear system (14) can be written with a block bordered diagonal (arrowhead) structure given by

$$\begin{bmatrix} W_1 & & & & A_1 \\ & W_2 & & & A_2 \\ & & W_3 & & A_3 \\ & & & \ddots & \vdots \\ & & & & W_{NS} & A_{NS} \\ A_1^T & A_2^T & A_3^T & \cdots & A_{NS}^T & \delta_1 I \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \\ \vdots \\ \Delta v_{NS} \\ \Delta d \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{NS} \\ r_d \end{bmatrix}, \quad (15)$$

where  $r_k^T = -[(\nabla_{x_k} \mathcal{L}_k^\ell)^T, (c_k^\ell)^T, (D_k x_k^\ell - \bar{D}_k d^\ell)^T]$ ,  $\Delta v_k^T = [\Delta x_k^T \Delta \lambda_k^T \Delta \sigma_k^T]$ ,  $A_k^T = [0 \ 0 \ -\bar{D}_k^T]$ ,

$$W_k = \begin{bmatrix} H_k^\ell + \delta_1 I & \nabla_{x_k} c_k^\ell & D_k^T \\ (\nabla_{x_k} c_k^\ell)^T & -\delta_2 I & 0 \\ D_k & 0 & -\delta_2 I \end{bmatrix}$$

for  $k = 1, \dots, NS$ , and  $r_d = \sum_{k=1}^{NS} \bar{D}_k^T \sigma_k^\ell$ .

The IPOPT algorithm requires the solution of the *augmented system* (15), at each iteration along with the determination of its inertia (the number of positive and negative eigenvalues).

The inertia calculation can be done efficiently with sparse, indefinite  $L^T B L$  solvers. Here, multiple factorizations of the *augmented system* are performed using different trial values of  $\delta_1$  and  $\delta_2$  until the proper inertia (i.e., #positive eigenvalues = #variables, #negative eigenvalues = #constraints) is obtained at each iteration. This guarantees that the Newton step obtained from (15) is a descent direction, as required by the filter line-search globalization strategy. Now, if values of  $\delta_1 = \delta_2 = 0$  give the proper inertia at the solution of the NLP (10), then it is possible to conclude that the solution satisfies strict second-order optimality conditions and has linearly independent constraints. As a result, the estimated parameters for this problem have unique values (Zavala and Biegler, 2006).

The linear system (15) can be solved, in principle, with any general direct linear solver configured with IPOPT. However, as the problem size grows, the time and memory requirements can make this approach intractable. Instead, applying a Schur complement decomposition allows an efficient parallel solution technique.

Eliminating each  $W_k$  from (15) we get the following expression for  $\Delta d$ :

$$\left[ \delta_1 I - \sum_{k=1}^{NS} A_k^T (W_k)^{-1} A_k \right] \Delta d = r_d - \sum_{k=1}^{NS} A_k^T (W_k)^{-1} r_k \quad (16)$$

which requires forming the Schur complement,  $B = \delta_1 I - \sum_{k=1}^{NS} A_k^T (W_k)^{-1} A_k$ , and solving this dense symmetric linear system for  $\Delta d$ . Once a value for  $\Delta d$  is known, the remaining variables can be found by solving the following system:

$$W_k \Delta v_k = r_k - A_k \Delta d \quad (17)$$

for each  $k = 1, \dots, NS$ . Note that in this strategy, the factorization of  $W_k$  and the solution of (17) can be performed independently in different processors. The Schur complement decomposition strategy applies specifically to the solution of the *augmented system* within the overall IPOPT algorithm and simply replaces the general default linear solver. The sequence of steps in the overall IPOPT algorithm is not altered, and as such, this specialized strategy inherits all of the convergence properties of the IPOPT algorithm for general nonlinear programs.

Furthermore, this decomposition strategy is straightforward to parallelize with excellent scaling properties. With  $M = \text{dim}(d)$ , the number of global parameters, the number of linear solves of the  $W_k$  blocks required by the decomposition approach is  $NS \cdot M + 2NS$ . If the number of available processors in a distributed cluster is equal to  $NS$  (one processor for each scenario), then the number of linear solves required by each processor is only  $M + 2$ , independent of the number of scenarios. This implies an approach that scales well with the number of scenarios. As we increase the number of scenarios under consideration, the cost of the linear solve remains fairly constant (with minimal communication overhead) as long as an additional processor is available for each new scenario. More importantly, the memory required on each processor is also nearly constant, allowing us to expand the number of scenarios and, using a large distributed cluster, move beyond the memory limitation of a standard single processor machine.

The efficient use of a distributed cluster to solve large problems that were previously not possible with a single standard machine is a major driving force of this work.

#### 4. Implementation of internal decomposition

The Schur complement algorithm described above is well known. Nevertheless, the implementation of this linear decomposition in most existing NLP software requires a nontrivial modification of the code. In many numerical codes, the particular data structures used for storing vectors and matrices are exposed to the fundamental algorithm code. With this design, it is straightforward to perform any necessary mathematical operations efficiently within the algorithm code. However, changing the underlying data representation (e.g. storing a vector in block form across a distributed cluster instead of storing it as a dense array) requires that the algorithm code be altered every place it has access to the individual elements of these vectors or matrices. Developing algorithms that exploit problem specific structure through internal decomposition requires the use of efficient (and possibly distributed) data structures that inherently represent the structure of the problem. In addition, it also requires the implementation of mathematical operations that can efficiently exploit this structure. If the fundamental algorithm code is intimately aware of the underlying data representation (primarily of vectors and matrices) then altering that representation for a particular problem structure can require a significant modification of the code.

In IPOPT 3.2, special care was taken to separate the fundamental algorithm code from the underlying data representations. The high-level structure of IPOPT is described in Fig. 1. The fundamental algorithm code communicates with the problem specification through an NLP interface. Moreover, the fundamental algorithm code is never allowed to access individual elements in vectors or matrices and is purposely unaware of the underlying data structures within these objects. It can perform only operations on these objects through various linear algebra interfaces. While the algorithm is independent of the underlying data structure, the NLP implementation needs to have access to the internal representation so it can fill the necessary data (e.g. specify the values of Jacobian entries). As a consequence, the NLP implementation is aware of the particular linear algebra implementation, but returns only interface pointers

to the fundamental algorithm code. The IPOPT package comes with a default linear algebra representation and a default set of NLP interfaces, but this design allows data representations and mathematical operations to be modified for a particular problem structure without changes to the fundamental algorithm code. Similar ideas have also been used in the design of reduced-space SQP codes, particularly for problems constrained by partial differential equations (Bartlett, 2001, 2002; Bartlett and van Bloemen Waanders, 2002).

In this work, we tested the redesigned IPOPT framework by implementing the Schur complement decomposition approach for the multi-scenario design problem. This implementation makes use of the message passing interface (MPI) to allow parallel execution on a distributed cluster. The implementation uses the composite design pattern and implements a composite NLP that forms the overall multi-scenario problem by combining individual NLP instances for each scenario. This implementation has also been interfaced to AMPL (Fourer et al., 1992), allowing the entire problem to be specified using individual AMPL NLP models for each scenario and AMPL suffixes to describe the connectivity, these implicitly define the  $D_k$ ,  $\bar{D}_k$ , and  $S_k$  matrices in (9). This allows the formulation of large multi-scenario problems with relative ease. Furthermore, when solving the problem in parallel, each processor only evaluates functions for its own scenarios, allowing distribution of data and parallelization of these computations across processors.

Parallel implementations for vectors and matrices have also been developed that distribute individual blocks across processors. All the necessary vector operations (e.g. BLAS operations, etc.) have been implemented for efficient calculation in parallel. Finally, a distributed solver has been written for the *augmented system* that uses a parallel version of the algorithm described in the previous section. This distributed solver uses a separate linear solver instance for the solution of each of the  $W_k$  blocks (and can use any of the linear solvers already interfaced with IPOPT). This separation allows solution of heterogeneous multi-scenario problems where the individual scenarios may have different structures. Finally, the distributed solver calls LAPACK routines for the dense linear solve of the Schur complement.

#### 5. LDPE case study

In this work, we apply the proposed decomposition strategy for the solution of large-scale parameter estimation problems arising in industrial LDPE reactors. LDPE grades are produced in high-pressure, multi-zone tubular reactors. Reacting in gas phase at high temperature (130–300 °C) and pressure (1500–3000 atm), ethylene is polymerized through a free-radical mechanism (Kiparissides et al., 2005) in the presence of complex mixtures of peroxide initiators. A typical tubular reactor can be described as a jacketed, multi-zone device with a predefined sequence of reaction and cooling zones. Different configurations of monomer and initiator mixtures enter in feed and multiple sidestreams, and are selected to maximize the reactor productivity and obtain the desired polymer properties. The total reactor length ranges between 0.5 and

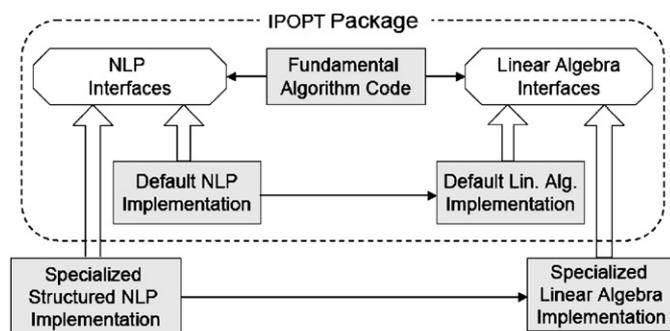


Fig. 1. Redesigned IPOPT structure, allowing for specialized linear algebra.

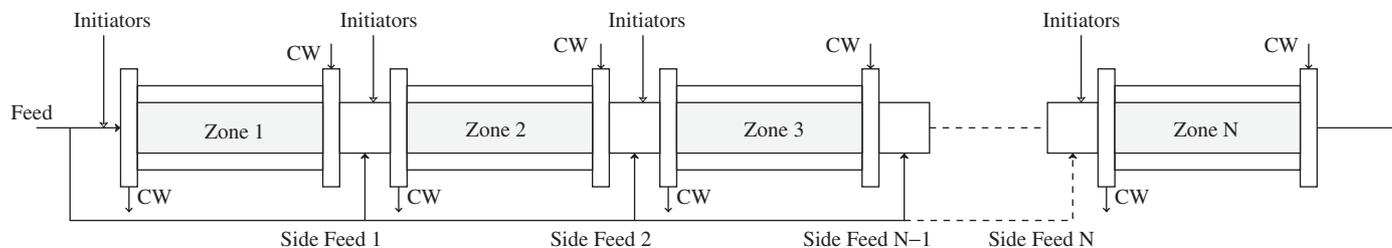


Fig. 2. Schematic representation of a typical high-pressure LDPE tubular reactor.

Table 1  
Ethylene homopolymerization kinetic mechanism

Initiator(s) decomposition	Incorporation of CTAs
$I_i \xrightarrow{\eta_i k_{di}} 2R, i = 1, N_I$	$P_r + S_i \xrightarrow{k_{spi}} P_{r+1}, i = 1, N_S$
Chain initiation	Termination by combination
$R \cdot + M \xrightarrow{k_{I1}} P_1$	$P_r + P_x \xrightarrow{k_{ic}} M_{r+x}$
Chain propagation	Termination by disproportionation
$P_r + M \xrightarrow{k_p} P_{r+1}$	$P_r + P_x \xrightarrow{k_{id}} M_r + M_x$
Chain transfer to monomer	Backbiting
$P_r + M \xrightarrow{k_{fm}} P_1 + M_r$	$P_r \xrightarrow{k_b} P_r'$
Chain transfer to polymer	$\beta$ -scission of sec- and tert-radicals
$P_r + M_x \xrightarrow{k_{fp}} P_x + M_r$	$P_r \xrightarrow{k_\beta} M_r^- + P_1$
Chain transfer to CTAs	
$P_r + S_i \xrightarrow{k_{si}} P_1 + M_r, i = 1, N_S$	

2 km, while its internal diameter does not exceed 70–80 mm. A schematic representation of a typical tubular reactor is presented in Fig. 2. The final end-use properties of the different LDPE grades are mainly correlated to the polymer density and macromolecular properties. Different additives or chain-transfer agents (CTAs) are added to the axial feed streams to control the polymer melt index. In general, the required polymer properties are enforced through complex recipes that try to keep the reactor at strict operating conditions.

There exist a number of comprehensive mathematical models for LDPE tubular reactors available in the literature. These models comprise detailed polymerization kinetic mechanisms and rigorous methods for the prediction of the reacting mixture thermodynamic and transport properties at extreme conditions. In this work, we consider a previously reported first-principles model describing the gas-phase free-radical homopolymerization of ethylene in the presence of several different initiators and CTAs at supercritical conditions (Zavala and Biegler, 2006). The mechanism postulated to describe the homopolymerization kinetics is presented in Table 1. Here, the symbols  $I_i$ ,  $i = 1, \dots, N_I$ ,  $R \cdot$ ,  $M$  and  $S_i$ ,  $i = 1, \dots, N_S$  denote the initiators, radicals, monomer, and CTA molecules, respectively. The symbol  $\eta_i$  represents the efficiency of initiator  $i$ ,  $P_r$  represents “live” polymer chains and  $M_r$  are “dead” polymer chains with  $r$  monomer units. The corresponding reaction rates for the monomers, initiators, CTAs and “live” and “dead” polymer chains can be obtained by combining the reaction rates of the elementary reactions describing their production and

consumption. Here, we recognize that a complete description of the polymer chain molecular weight distributions requires an extremely large number of population balances for the polymer chains. To avoid this, the method of moments (Ray, 1972) is used to describe macromolecular properties of the copolymer. The method of moments is based on the statistical representation of the polymer average molecular weights in terms of the leading moments of the number chain-length distributions of the “live” and “dead” polymer chains. In this model, the number chain-length distributions for  $P_r$  and  $M_r$  are considered. Accordingly, the moments of the number chain-length distributions are defined as

$$\lambda_n = \sum_{r=1}^{\infty} r^n R(r), \quad n = 0, 1, 2, \quad (18)$$

$$\mu_n = \sum_{r=1}^{\infty} r^n D(r), \quad n = 0, 1, 2, \quad (19)$$

where  $R(r) = [P_r]$  and  $D(r) = [M_r]$ . With this, macromolecular properties of the polymer can be obtained in terms of the leading moments of the chain-length distributions. For instance, the polymer number- and weight-average molecular weights and polydispersity are given by

$$MW_n = MW_0 \frac{\mu_1}{\mu_0}, \quad (20)$$

$$MW_w = MW_0 \frac{\mu_2}{\mu_1}, \quad (21)$$

$$PDI = \frac{MW_w}{MW_n}, \quad (22)$$

where  $MW_0$  is the average molecular weight of a building unit in the polymer chain. The number of short- and long-chain branches per 1000 atoms can be obtained from

$$LCB = 500 \frac{C_{LCB}}{\mu_1}, \quad (23)$$

$$SCB = 500 \frac{C_{SCB}}{\mu_1}, \quad (24)$$

where expressions for the calculation of the concentration  $C_{LCB}$  and  $C_{SCB}$  can be derived from the kinetic mechanism. Despite the fact that all these macromolecular properties provide a relatively accurate description of the structural properties of the polymer, they are rarely used to monitor the quality of the polymer in industrial reactors since they are difficult to measure

online. Instead, the polymer melt index and density are normally used as raw quality measures. The polymer density  $\rho_{\text{pol}}$  is correlated to the number of short-chain branches

$$\rho_{\text{pol}} = a_0 + a_1 \text{SCB} \quad (25)$$

and the polymer melt index can be correlated to the rest of the macromolecular properties as

$$\log_{10}(\text{MI}) = a_2 + a_3 \log_{10}(\text{MW}_w) + a_4 \log_{10}(\text{PDI}) + a_5 \log_{10}(\text{LCB}). \quad (26)$$

The complexity of the rigorous model is often reduced by making the following validated assumptions:

- the reacting mixture forms a single supercritical phase;
- plug flow is observed along the reactor;
- net production rates of the radicals and “live” polymer chains are negligible (quasi-steady-state assumption) (Kiparissides et al., 1993).

Considering this, it is possible to derive sets of *steady-state* differential molar and energy balances describing the evolution of the reacting mixture along each reactor zone. The detailed design equations are reported in Zavala and Biegler (2006).

Perhaps the most difficult problem in simulating the operation of high-pressure LDPE reactors is the selection of appropriate values for the kinetic rate parameters in Table 1. The rate constants have the general Arrhenius form

$$k_i = k_i^0 \exp \left[ -\frac{\Delta E_{a_i} + P \Delta E_{v_i}}{RT} \right], \quad (27)$$

where subindex  $i$  belongs to the entire set of elementary reactions in the kinetic mechanism,  $k_i^0$  denotes the pre-exponential factor,  $\Delta E_{a_i}$  the activation energy,  $\Delta E_{v_i}$  the activation volume,  $T$  is the temperature,  $P$  is the pressure and  $R$  is the universal gas constant. Despite the importance of the kinetic parameters, there is not a consistent set of values that can be obtained from literature reports. Furthermore, even if the parameter values could be obtained from idealized laboratory conditions, they will not be in general applicable to the full-scale rigorous reactor model, due to natural uncertain disturbances arising in industrial units such as complex kinetic-transport interactions (Yoon et al., 2004). As a result, it is difficult to find a reliable set of parameters; instead, the parameters need to be tuned using the full rigorous model to match industrial reactor data directly.

In addition to limited kinetic parameters data, LDPE tubular reactors are also subject to persistent variability over the operating horizon. This requires the selection and on-line estimation of adjustable parameters to account for this variability. One of the most fundamental and complex problems associated to the operation of LDPE tubular reactors is the severe and random fouling of the inner reactor wall due to a continuous polymer build-up. This phenomenon is difficult to predict by means of simple mechanistic models (Buchelli et al., 2005). The most simple engineering way to handle this problem is to estimate the heat-transfer coefficient (HTC) associated to each reactor zone to match

the reactor temperature profile (Kiparissides et al., 1996). In addition, there exists uncertainty associated to the decomposition mechanism of the initiator mixtures along the reactor. These initiators decompose to generate the radicals that start the polymerization. At each feed point, a typical initiator mixture can include up to four different initiators with different chemical properties. The initiator decomposition reactions include sets of complex reaction subnetworks involving the formation of highly active intermediate species that can react among each other or with impurities in the reacting mixture before generating the desired radicals. Thus, there is an efficiency factor  $\eta_i$  associated to the decomposition of each initiator. These initiator efficiencies are strongly dependent on a large number of factors such as the reacting mixture temperature and pressure, the degree of mixing at the feed points and the presence of other species such as impurities or CTAs. Moreover, the efficiency of an individual initiator might vary with its concentration in the reacting mixture (Luft et al., 1977; Seidl and Luft, 1981). In LDPE tubular reactors, wide variations of the reacting mixture temperature, pressure, composition and physical properties are observed. As a consequence, wide variations of the efficiencies are expected as well along the reactor and over time due to the accumulation of impurities. To cope with this, the initiator efficiencies can be estimated for each reaction zone in order to match the plant reactor temperature profile. Previous studies have shown satisfactory results using this technique; this approach is followed in this work.

To illustrate the accuracy of the described rigorous model, a comparison between plant and predicted temperature profiles for an industrial reactor is presented in Fig. 3. The predicted profile was obtained using estimated HTCs and initiator efficiencies that match the model temperature profile to the plant measurements at a particular point in time.

According to the above description, the steady-state evolution of the reacting mixture along the multiple reactor zones can be formulated as a multi-stage DAE system of the form

$$\begin{aligned} \mathbf{F}_{k,j} \left[ \frac{dz_{k,j}(t)}{dt}, z_{k,j}(t), y_{k,j}(t), p_{k,j}, \Pi \right] &= 0, \\ \mathbf{G}_{k,j} [z_{k,j}(t), y_{k,j}(t), p_{k,j}, \Pi] &= 0, \\ z_{k,j}(0) &= \phi(z_{k,j-1}(t_{L_{k,j-1}}), u_{k,j}), \\ k &= 1, \dots, \text{NS}, \quad j = 1, \dots, \text{NZ}_k. \end{aligned} \quad (28)$$

Notice the appearance of subindex  $j$  denoting a particular stage or reactor zone defined for a particular operating scenario  $k$  and  $\text{NZ}_k$  is the total number of zones for the reactor in scenario  $k$ , so this formulation allows estimation over different reactor configurations. In addition, note that the zone DAE models are coupled through material and energy balances  $\phi(\cdot)$  at the feed points where the input variables  $u_{k,j}$  include the flow rates and temperatures for the monomer, initiator, CTA and cooling water side streams along the reactor and  $t_{L_{k,j}}$  denotes the total length of zone  $j$  in scenario  $k$ . Symbol  $p_{k,j}$  denotes local parameters corresponding to the HTC and multiple initiator efficiencies at each zone  $j$  and scenario  $k$  and  $\Pi$  corresponds to the kinetic rate constants. The reactor model contains around 130 ordinary differential equations and 500 algebraic equations

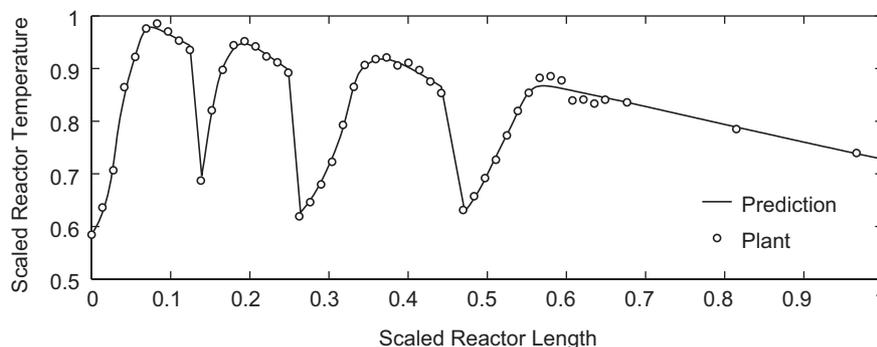


Fig. 3. Predicted and plant temperature profiles for a typical LDPE tubular reactor.

for each instance  $k$ . That is, the total number of equations in (28) increases linearly with the number of scenarios  $k$ . In addition to the large number of equations, the reactor model is computationally expensive to solve due to the high nonlinearity and stiffness of the DAEs and the high degrees of algebraic coupling and parametric sensitivity.

Once the model has been defined, the objective is to estimate the kinetic parameters to match the plant reactor operating conditions and polymer properties. However, due to the uncertainty associated to the fouling and initiator decomposition mechanisms, it is also necessary to include the HTCs and initiator efficiencies in the set of estimated parameters. In order to capture the interaction of  $p_{k,j}$  and  $\Pi$  and to account for the measurement errors in the multiple of flow rates, concentrations, temperatures and pressures around the reactor, we consider multi-scenario EVM estimation problems of the form

$$\begin{aligned} \min_{\Pi, p_{k,j}, u_{k,j}} & \sum_{k=1}^{NS} \sum_{j=1}^{NZ_k} \sum_{i=1}^{NM_{k,j}} (y_{k,j}(t_i) - \bar{y}_{k,j,i})^T \\ & \times \mathbf{V}_y^{-1}(y_{k,j}(t_i) - \bar{y}_{k,j,i}) \\ & + \sum_{k=1}^{NS} \sum_{j=1}^{NZ_k} (u_{k,j} - u_{k,j}^M)^T \mathbf{V}_u^{-1}(u_{k,j} - u_{k,j}^M) \\ \text{s.t.} & \mathbf{F}_{k,j} \left[ \frac{dz_{k,j}(t)}{dt}, z_{k,j}(t), y_{k,j}(t), u_{k,j}, p_{k,j}, \Pi \right] = 0, \\ & \mathbf{G}_{k,j} [z_{k,j}(t), y_{k,j}(t), u_{k,j}, p_{k,j}, \Pi] = 0, \\ & \mathbf{H}_{k,j} [z_{k,j}(t), y_{k,j}(t), p_{k,j}, \Pi] \leq 0, \\ & z_{k,j}(0) = \phi(z_{k,j-1}(t_{L_{k,j-1}}), u_{k,j}), \\ & k = 1, \dots, NS, \quad j = 1, \dots, NZ_k, \end{aligned} \quad (29)$$

where the output variables contain the reactor temperature profile along each zone, jacket inlet and outlet temperatures at each zone, macromolecular properties ( $MW_w$ ,  $MW_n$ , LCB) and quality properties (MI,  $\rho_{pol}$ ) at the reactor outlet which are matched to the corresponding available plant measurements for each operating scenario or data set  $k$ .

As reported in our previous work, it is possible to obtain reliable parameters and improve the predictive capabilities of

LDPE reactor models through the application of systematic strategies for parameter estimation (Zavala and Biegler, 2006). A simultaneous approach was applied to the solution of (29) and the resulting NLP of the form in (7) was solved on a single processor machine using IPOPT without internal decomposition. Following this approach, it was shown that the size of the resulting parameter 95% confidence regions can be reduced substantially as the number of data sets is increased. That is, the reliability of the estimated parameters can be improved and conclusions on model structure limitations can be drawn. On the other hand, it was also shown that the associated NLPs become quickly intractable due to the size and complexity of the LDPE reactor model. As expected, the key bottleneck was the factorization of the KKT matrix which could only be performed for NLPs with a small number of data sets.

In this work, we apply the described parallel decomposition strategy to the NLPs resulting from the application of a simultaneous approach to (29). As opposed to the strategy from Zavala and Biegler (2006), we now exploit the natural multi-scenario structure in (10). Since the decomposition strategy avoids the factorization of the full KKT matrix in a single processor, it is able to avoid memory bottlenecks and handle a large number of data sets in the estimation problem, thus improving the reliability of the estimated parameters.

To formulate the estimation problem (29) as a multi-scenario NLP (10), we perform a full discretization of the differential and algebraic variables and group the resulting set of variables by data sets or scenarios  $k$ . For each data set, we use a total of 16 finite elements for the reaction zones, two finite elements for the cooling zones and three collocation points for the discretization in (29), giving rise to NS *individual* NLP instances, each one with around 12,000 constraints and 92 degrees of freedom, from which 32 correspond to the local parameters  $p_k$ , 25 to global parameters  $\Pi$  and 35 to the input variables  $u_k$ . In order to obtain exact first and second derivative information, the NLP instances are implemented as NS separate AMPL models that internally *indicate* the set of variables corresponding to the global parameters  $\Pi$ . This is required to build the linking variables vector in IPOPT.

In Fig. 4 we present computational results associated to the solution of multi-scenario NLPs with up to 32 data sets. The results were obtained in a Beowulf type cluster using standard

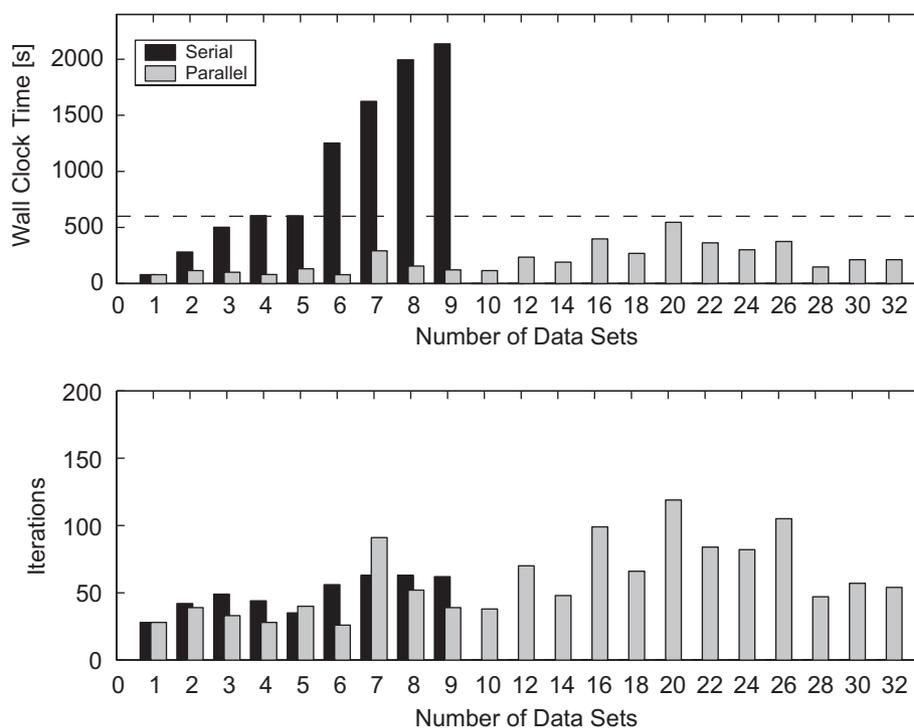


Fig. 4. Total time and number of iterations for the solution of multi-scenario NLPs with IPOPT. Serial and parallel implementations.

Intel Pentium IV Xeon 2.4 GHz, 2 Gb RAM processors running on Linux. The parallel results are compared against those obtained from the serial solution of the multi-scenario problems in a single processor with similar characteristics. As seen in the figure, the serial solution of the multi-scenario NLPs exhausts the available memory when the number of data sets exceeds nine, while the parallel implementation overcomes this memory bottleneck and solves problems with over 32 data sets. In all of the analyzed cases, no regularization of the KKT matrix was required at the solution (no inertia correction). The information provided in the estimation problems is sufficient to estimate the full set of parameters uniquely. Here, the largest problem solved contains around 4100 differential and 16,000 algebraic equations and 2100 degrees of freedom. Moreover, notice that the solution time significantly increases in the serial implementation as we add more data sets. The solution of the nine data set problem takes more than 30 min. In contrast, the parallel solution takes consistently less than 10 min regardless of the number of data sets. Finally, note that the solution times and number of iterations do not follow any particular trend, presenting “random” jumps as we add or remove data sets. It is important to emphasize that this behavior is problem (and data) dependent. In fact, the solution of the 32 data set problem requires fewer iterations than that with 20 data sets. This behavior is mainly attributed to the nonlinearity of the constraints, the influence of the initialization with different NS, and ill-conditioning of the KKT matrix. Moreover, the high nonlinearity of the constraints gives rise to directions of negative curvature, which require additional inertia correction steps (more factorizations of linear system (15) per iteration); this is entirely problem dependent.

Motivated by this behavior, we present in Fig. 5 computational results for the time required per iteration and factorization of the KKT matrix; this is a more consistent measure of the scalability of the proposed strategy. For the parallel approach, notice that the effect of parallelism is reflected not in the time required per iteration but on the time per factorization. Nevertheless, the time per iteration can be consistently kept below 5 s, while the factorization in the serial approach can take as much as 35 s before running out of memory.

In Table 2 we present a summary of the computational results for both the sequential and parallel approaches. From the serial and parallel results, it is possible to observe that the number of iterations taken by the IP algorithm is hardly affected as we add degrees of freedom to the estimation problem. It is important to emphasize that this desirable property can be obtained with full-space solvers using *exact* derivative information. As the proposed parallel decomposition strategy does not alter the core NLP algorithm in IPOPT, these desirable properties are retained.

## 6. Conclusions and future work

In this work, a decomposition strategy is proposed for parallel solution of large-scale parameter estimation problems. These multi-scenario problems are solved following an IP strategy that allows to exploit the resulting block bordered diagonal structure of the KKT matrix. As a result, the decomposition approach occurs only at the linear algebra level, preserving the convergence properties of the IP method. The strategy has been implemented within IPOPT 3.2, a recently redesigned IP algorithm that allows for the implementation of specialized,

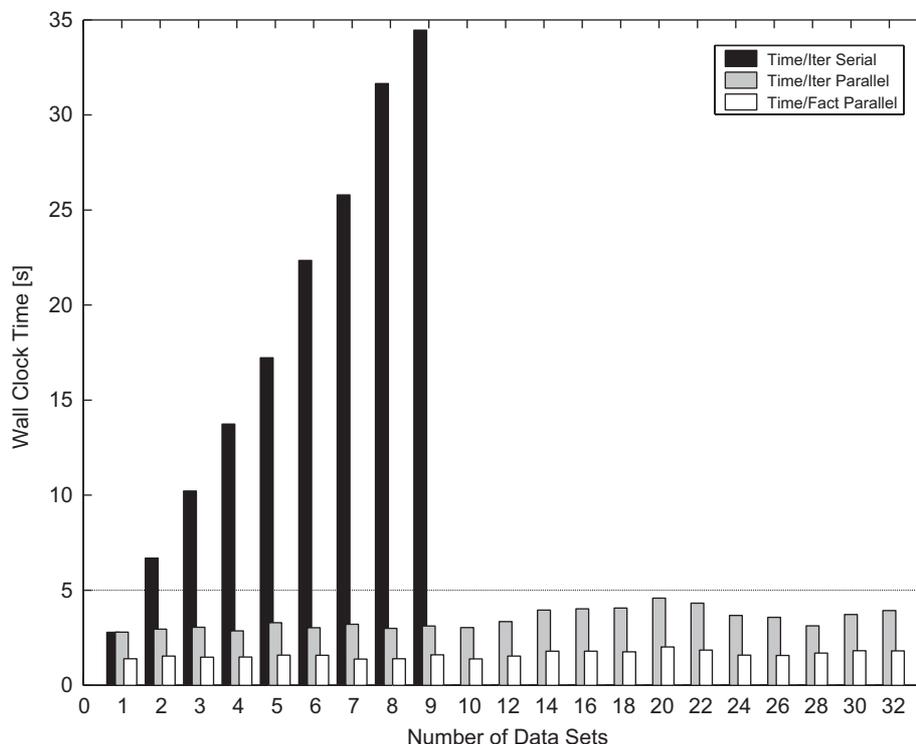


Fig. 5. Time per iteration and per factorization of the KKT matrix during the solution of multi-scenario NLPs with IPOPT. Serial and parallel implementations.

Table 2

Summary of computational results associated to the solution of multi-scenario NLPs with IPOPT

NS	NLP statistics				Serial			Parallel			
	$m$	DOF	LB	UB	It	$\theta_{\text{total}}$ (s)	$\theta_{it}$ (s)	It	$\theta_{\text{total}}$ (s)	$\theta_{it}$ (s)	$\theta_{kkt}$ (s)
1	12 319	92	425	412	28	78.24	2.79	28	78.24	2.79	1.40
2	24 638	159	850	824	42	280.86	6.69	39	115.22	2.95	1.54
3	36 957	226	1275	1236	49	500.75	10.22	33	100.62	3.05	1.48
4	49 276	293	1700	1648	44	604.45	13.74	28	80.08	2.86	1.48
5	61 643	361	2126	2061	35	603.20	17.23	40	131.76	3.29	1.59
6	73 962	428	2551	2473	56	1251.87	22.35	26	78.77	3.03	1.58
7	86 953	495	2976	2885	63	1624.59	25.79	91	291.91	3.21	1.38
8	99 944	562	3401	3297	63	1994.17	31.65	52	155.90	3.00	1.39
9	1 12 935	629	3826	3709	62	2136.82	34.46	39	121.55	3.12	1.60
10	1 25 254	696	4251	4121				38	115.34	3.04	1.39
12	1 37 573	763	4676	4533				70	234.50	3.35	1.53
14	1 50 564	830	5101	4945				48	189.94	3.96	1.79
16	2 00 512	1098	6801	6593				99	398.35	4.02	1.79
18	2 25 822	1232	7651	7417				66	268.03	4.06	1.76
20	2 51 132	1366	8501	8241				119	545.19	4.58	2.02
22	2 75 098	1500	9351	9065				84	363.14	4.32	1.85
24	3 00 408	1634	10 201	9889				82	301.13	3.67	1.58
26	3 25 046	1768	11 051	10 713				105	375.00	3.57	1.57
28	3 49 684	1902	11 901	11 537				47	147.14	3.13	1.69
30	3 74 994	2036	12 751	12 361				57	212.31	3.72	1.81
32	3 99 632	2170	13 601	13 185				54	212.15	3.93	1.81

$m$ , number of constraints; DOF, number of degrees of freedom; LB, number of lower bounds; UB, number of upper bounds; It, number of iterations;  $\theta_{\text{total}}$ , total wall clock time;  $\theta_{it}$ , wall clock time per iteration;  $\theta_{kkt}$ , wall clock time per factorization of KKT matrix.

structure-exploiting linear algebra strategies. The proposed approach is suitable for parallel computing architectures and has been used for the solution of large-scale parameter estimation

problems arising in industrial LPDE tubular reactors. Using this case study, it is shown that multi-set DAE-constrained estimation problems can be solved quickly and efficiently with

