Robust Simulation and Design Using Parametric Interval Methods

Matthew D. Stuber* and Paul I. Barton†
Dept. of Chemical Engineering, Massachusetts Institute of Technology, Cambridge MA, USA

Abstract. A method is presented for guaranteeing robust steady-state operation of chemical processes using a model-based approach, taking into account uncertainty in the model parameters and disturbances in the process inputs. Intractable bilevel optimization formulations have been proposed for this problem in the past. A new approach is presented in which the equality constraints (process model equations) are solved numerically for the process variables as implicit functions of the uncertainty parameters and controls. The problem is then formulated as a semi-infinite program (SIP) constrained only by the performance specifications as semi-infinite inequality constraints. A rigorous algorithm for solving such SIPs is proposed, making no assumptions on convexity, which makes use of the novel developments of parametric interval Newton methods for bounding implicit functions and McCormick relaxations of algorithms. Upper and lower bounding techniques are applied within the Branch & Bound framework. Finite ϵ-optimal convergence to the global solution of the SIP is guaranteed with the existence of a Slater point arbitrarily close to a maximizer.

Keywords: Interval Analysis, Semi-infinite Optimization, Global Optimization, McCormick Relaxation, Robust Simulation, Design Under Uncertainty.

1. Introduction

The task of a process design engineer is to design a process system that will meet all predetermined performance specifications given limited information and resources. Imprecise data and lack of complete environmental information, among other sources, introduce various uncertainties that must be accounted for in the design. In applications such as subsea oil and gas production, increasingly extreme environments and costs make building physical pilot plant systems implausible. In this case, a model-based approach is more plausible. The first question a design engineer must address becomes: "Given a process model, and taking into account uncertainty in the model and disturbances to the inputs of the system, do there exist control settings such that, at steady state, the physical system will always meet performance and/or safety specifications?" The primary goal is thus to be able to give robustness guarantees for the performance of physical process systems, at the design stage, using a model-based approach.

The robustness problem can be formulated as the following bilevel optimization problem (Haleman and Grossman, 1983):

* Email: stuber@mit.edu
† To whom all correspondence should be addressed, Email: pib@mit.edu

4th International Workshop on Reliable Engineering Computing (REC 2010)
Edited by Michael Beer, Rafi L. Muhanna and Robert L. Mullen
Copyright © 2010 Professional Activities Centre, National University of Singapore.
ISBN: 978-981-08-5118-7. *Published by* Research Publishing Services.
doi:10.3850/978-981-08-5118-7_003

$$\begin{aligned}\eta^* = & \max_{(\mathbf{d},\mathbf{p})\in D\times P,\eta\in\mathbb{R}} \eta \\ \text{s.t.}\ \ & \eta \le \min_{\mathbf{u}\in U,\mathbf{x}\in X} g(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{p}) \\ & \qquad \text{s.t.}\ \ \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{p}) = \mathbf{0}\end{aligned} \tag{1}$$

where $\mathbf{h} : X \times U \times D \times P \to \mathbb{R}^{n_x}$ and $g : X \times U \times D \times P \to \mathbb{R}$ are the steady-state model equations and performance specification, respectively, with $X \subset \mathbb{R}^{n_x}, U \subset \mathbb{R}^{n_u}, D \subset \mathbb{R}^{n_d}, P \subset \mathbb{R}^{n_p}$. Here, $\mathbf{x} \in X$ will be the process state variables, $\mathbf{d} \in D$ are the disturbance uncertainty parameters, $\mathbf{p} \in P$ are the model uncertainty parameters, and $\mathbf{u} \in U$ are the control variables that can be adjusted in response to disturbances. Upon solving (1), if $\eta^* \le 0$, the design is said to be robust. That is, the controls can be adjusted to guarantee the performance specification is satisfied ($g(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{p}) \le 0$), for all possible uncertainty realizations. However, due to the complexity of typical robustness problems, this bilevel formulation is difficult to solve or simply not tractable. Likewise, the exponential worst-case runtime of all known deterministic global optimization algorithms motivates the need for a new, reduced-space, formulation.

Assuming differentiability of $\mathbf{h}$ and a nonsingular Jacobian matrix with respect to $\mathbf{x}$ on an open set $Q \subset \mathbb{R}^{n_x}$, the process model equations can be solved for $\mathbf{x}$ as:

$$\mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{p}) = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{x}(\mathbf{u},\mathbf{d},\mathbf{p}), \tag{2}$$

by asserting the Implicit Function Theorem. By representing the internal state variables $\mathbf{x}$ as implicit functions of the controls and uncertainty parameters, there is a significant reduction in size of the optimization problem. Likewise, by reformulating the inner program as

$$g(\mathbf{x}(\mathbf{u},\mathbf{d},\mathbf{p}),\mathbf{u},\mathbf{d},\mathbf{p}) \ge \eta,\ \forall \mathbf{u} \in U,$$

the problem becomes a single-level optimization problem with a finite number of decision variables subject to an infinite number of constraints, also known as a semi-infinite program (SIP). If these two techniques are applied, the bilevel program (1) can be reformulated as:

$$\begin{aligned}\eta^* = & \max_{(\mathbf{d},\mathbf{p})\in D\times P,\eta\in\mathbb{R}} \eta \\ \text{s.t.}\ \ & \eta \le g(\mathbf{x}(\mathbf{u},\mathbf{d},\mathbf{p}),\mathbf{u},\mathbf{d},\mathbf{p}),\ \ \forall \mathbf{u} \in U.\end{aligned} \tag{3}$$

The key difference from regular SIPs, in general, is that the semi-infinite constraint function g is implicitly defined and therefore not known explicitly. The primary focus of this paper will be on solving SIPs with embedded implicit functions of the form (3), that arise in robust optimization applications. However, the tools and results summarized in this paper apply, in general, to any SIP where the semi-infinite constraint is implicitly defined.

2. Background

This section provides the reader with an overview of the background mathematical concepts used in the proposed algorithm for solving (3). Let $\Xi \subset \mathbb{R}^{n_x}$, $\Omega \subset \mathbb{R}^{n_u}$, $\Delta \subset \mathbb{R}^{n_d}$, $\Pi \subset \mathbb{R}^{n_p}$ be open sets with $X \subset \Xi$, $U \subset \Omega$, $D \subset \Delta$, $P \subset \Pi$ as intervals. It is assumed that for all considered models, $\mathbf{h} : \Xi \times \Omega \times \Delta \times \Pi \to \mathbb{R}^{n_x}$ is continuously differentiable.

2.1. Interval Analysis

The algorithm for solving SIPs constrained by implicit functions as in (3), relies on interval analysis to generate valid enclosures of the range of implicit functions over the variable domain. This section will present the definitions and nomenclature of interval analysis in a manner attempting to preserve generality. For a more comprehensive discussion of interval methods, the reader is directed to (Moore, 1979) and (Neumaier, 1990)

Definition 2.1. *An interval $Z \subset \mathbb{R}^m$ is defined as the compact set:*

$$Z = \{\mathbf{z} \in \mathbb{R}^m : \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U\},$$

with $\mathbf{z}^L \in \mathbb{R}^m$ and $\mathbf{z}^U \in \mathbb{R}^m$ as the lower and upper bounds of the interval Z.

Definition 2.2. *The set of all real intervals is denoted $\mathbb{IR}$.*

Definition 2.3. *An interval $Z \in \mathbb{IR}^m$ is defined as an m-dimensional vector whose elements are intervals denoted by a subscript Z_i for $i = 1, \ldots, m$.*

Proposition 2.1. *An interval $Z \in \mathbb{IR}^m$ can be composed from n subintervals, $Z^i \subset Z$, such that $Z = \bigcup_{i=1}^{n} Z^i$.*

Definition 2.4 (Midpoint)**.** *The* midpoint *or median, of an interval $Z \in \mathbb{IR}^m$ will be defined as the vector of the midpoints of the elements:*

$$m(Z_i) \equiv \frac{z_i^L + z_i^U}{2}, \quad i = 1, \ldots, m. \tag{4}$$

Definition 2.5 (Image)**.** *The* image *of the set $Z \in \mathbb{IR}^m$ under the mapping $\mathbf{f} : A \to \mathbb{R}^n$, with $Z \subset A$ is denoted as $\hat{\mathbf{f}}(Z)$ and has the upper and lower bounds $\hat{\mathbf{f}}^U(Z)$ and $\hat{\mathbf{f}}^L(Z)$, respectively.*

Definition 2.6 (Interval Hull)**.** *The* interval hull *of the image of a set $Z \in \mathbb{IR}^m$ under the mapping $\mathbf{f} : A \to \mathbb{R}^n$, with $Z \subset A$, is the smallest interval that encloses the image, and is denoted as $\hat{\mathbf{f}}^H(Z) = \left[\hat{\mathbf{f}}^L(Z), \hat{\mathbf{f}}^U(Z)\right]$.*

Definition 2.7. *An interval-valued function $F : A \to \mathbb{IR}^n$, evaluated at any $Z \subset A$, with $Z, A \in \mathbb{IR}^m$, is denoted as $F(Z)$.*

Definition 2.8 (Interval Extension)**.** *An interval-valued function $F : A \to \mathbb{IR}^n$, with $A \in \mathbb{IR}^m$, is called an* interval extension *of the real-valued function $\mathbf{f} : A \to \mathbb{R}^n$ on A, if*

$$F([\mathbf{z}, \mathbf{z}]) = \mathbf{f}(\mathbf{z}), \quad \forall \mathbf{z} \in A.$$

Definition 2.9. *An interval extension, $F(Z)$, of the function $\mathbf{f} : A \to \mathbb{R}^n$, at $Z \subset A$, with $Z, A \in \mathbb{IR}^m$, is called* exact *at Z, if $F(Z) = \hat{\mathbf{f}}^H(Z)$.*

Definition 2.10 (Inclusion Monotonicity (Moore, 1979; Neumaier, 1990)). *Let $A, B, Z \in \mathbb{IR}^m$. The interval-valued function $F : A \to \mathbb{IR}^n$ is called* inclusion monotonic *if for every $B, Z \subset A$,*

$$B \subset Z \Rightarrow F(B) \subset F(Z). \tag{5}$$

Definition 2.11 (Nested Sequences). *A sequence of intervals $\{Z^k\}$ is said to be* nested *if $Z^{k+1} \subset Z^k$ for every k.*

Definition 2.12 (Inclusion Function). *An interval-valued function $F : A \in \mathbb{IR}^m \to \mathbb{IR}^n$ is called an* inclusion function *of $\mathbf{f}$ on A if*

$$\mathbf{f}(\mathbf{z}) \in F(Z), \quad \forall \mathbf{z} \in Z,$$

for $Z \subset A$.

Theorem 2.1 ((Moore, 1979; Neumaier, 1990)). *Let $A \in \mathbb{IR}^m$. If $F : A \to \mathbb{IR}^n$ is an inclusion monotonic interval extension of $\mathbf{f}$ on A, then*

$$\hat{\mathbf{f}}(Z) \subset F(Z)$$

holds for every $Z \in \mathbb{IR}^m$ such that $Z \subset A$.

Proof. Proof can be found in (Moore, 1979) page 21. □

Definition 2.13 (Interval Width). *The* width *of an interval $Z \in \mathbb{IR}$ is defined as the distance between its upper and lower bounds:*

$$w(Z) = z^U - z^L.$$

Definition 2.14 (Excess Width (Moore, 1979)). *Let F be an inclusion monotonic interval extension of $\mathbf{f}$ on Z. Then $w(E(Z))$ is called the* excess width *of $F(Z)$, such that $w(F(Z)) = w(\hat{\mathbf{f}}(Z)) + w(E(Z))$.*

Definition 2.15. *An interval extension of the partial derivative of the continuously-differentiable function $f_i : A \subset \mathbb{R}^m \to \mathbb{R}$ with respect to z_j is denoted as*

$$F'_{i,j}(Z) \equiv \frac{\partial F_i}{\partial z_j}(Z)$$

for $Z \in \mathbb{IR}^m$ such that $Z \subset A$.

Definition 2.16. *An inclusion monotonic interval extension of the Jacobian matrix, $\mathbf{J}$, of a vector-valued function $\mathbf{f}$, on $Z \in \mathbb{IR}^m$ is denoted as*

$$J_{\mathbf{z}}(Z) \equiv \begin{bmatrix} \frac{\partial F_1}{\partial z_1}(Z) & \cdots & \frac{\partial F_1}{\partial z_n}(Z) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial z_1}(Z) & \cdots & \frac{\partial F_n}{\partial z_n}(Z) \end{bmatrix}. \tag{6}$$

2.2. Parametric Interval Newton-Type Methods

Parametric interval Newton-type methods are the primary tools for generating valid enclosures of implicit functions, which are required for solving (3). The theory, as well as a detailed discussion of the useful properties of parametric interval Newton methods, are only summarized here. One useful similarity between all the parametric interval Newton-type methods is that they define nested sequences of intervals. Define $\mathbf{y} = (\mathbf{d}, \mathbf{p})$ to be the general uncertainty variable and let $Y = D \times P$ represent the general uncertainty set such that $\mathbf{y} \in Y$. For ease of presentation, it will be assumed that an interval X^0 is known that contains the value of the implicit function $\mathbf{x}(\mathbf{u}, \mathbf{y})$, for every $(\mathbf{u}, \mathbf{y}) \in U \times Y$. Under this assumption, the basic application of the methods is guaranteed to converge to some interval $X^* \subset X^0$ that is also guaranteed to enclose the implicit solution. However, this assumption is not necessary for convergence to an interval enclosure and the general case is discussed in a paper currently in preparation.

Definition 2.17 (Parametric Interval Newton Method)**.** *Let $X^0 \in \mathbb{IR}^{n_x}$ be the initial guess interval with $X^0 \subset Q$. Let $H(\mathbf{x}, U, Y)$ be an inclusion monotonic interval extension of $\mathbf{h} : X^0 \times U \times Y \to \mathbb{R}^{n_x}$ on $U \times Y$ for every $\mathbf{x} \in X^0$. The parametric interval Newton method is then defined as:*

$$\begin{aligned} \mathbf{x}^k &:= m(X^k) \\ N\left(\mathbf{x}^k, X^k, U, Y\right) &:= \mathbf{x}^k - J_{\mathbf{x}}(X^k, U, Y)^{-1} H(\mathbf{x}^k, U, Y) \\ X^{k+1} &:= X^k \cap N\left(\mathbf{x}^k, X^k, U, Y\right). \end{aligned} \tag{7}$$

Since $X^0 \subset Q$, $J_{\mathbf{x}}(X^0, U, Y)$ does not contain any singular matrices, and therefore $J_{\mathbf{x}}(X^k, U, Y)$ also does not contain any singular matrices for every $X^k \subset X^0$. It should be noted that $\mathbf{x}^k$ does not need to be the midpoint of X^k but can be any point $\mathbf{x} \in X^k$. However, the midpoint is convenient and may offer some useful properties to the interval Newton-type algorithms (Hansen and Walster, 2004; Moore, 1979; Neumaier, 1990). For better convergence and enclosure properties, a sequential componentwise calculation of N is recommended. This technique is known as the *Gauss-Seidel* method and it is a result of arranging the parametric interval Newton operator, N, into the linear system

$$J_{\mathbf{x}}(X, U, Y)(N(\mathbf{x}, X, U, Y) - \mathbf{x}) = -H(\mathbf{x}, U, Y), \tag{8}$$

and solving for N componentwise. It is also important to mention that preconditioning (8) by some matrix $\mathbf{\Psi}$ is recommended to give better convergence and enclosure properties (Kearfott, 1990; Kearfott, 1996; Neumaier, 1990). It is common that $\mathbf{\Psi}$ is taken to be an approximate inverse of the midpoint of the interval Jacobian matrix J. The Gauss-Seidel method is defined in the following.

Definition 2.18 (Parametric Interval Newton w/ Gauss-Seidel)**.** *For $i = 1, 2, \ldots, n_x$:*

$$\begin{aligned} N_i^k &:= x_i^k - \left[b_i^k + \sum_{j=1}^{i-1} A_{i,j}^k \left(X_j^{k+1} - x_j^{k+1} \right) + \sum_{j=i+1}^{n_x} A_{i,j}^k \left(X_j^k - x_j^k \right) \right] / A_{i,i}^k \\ X_i^{k+1} &:= N_i^k \cap X_i^k, \end{aligned} \tag{9}$$

with $\mathbf{A}^k \in \mathbb{IR}^{n_x \times n_x}$ as $\mathbf{A}^k = \mathbf{\Psi}^k J_{\mathbf{x}}(X^k, U, Y)$, and $\mathbf{b}^k \in \mathbb{R}^{n_x}$ as $\mathbf{b}^k = \mathbf{\Psi}^k H(\mathbf{x}^k, U, Y)$.

Another method that does not require the inversion of an interval matrix, or division by intervals, such as the parametric interval Newton method, is known as the parametric Krawczyk method. It is defined in the following.

Definition 2.19 (Parametric Krawczyk Method). *Let $X^0 \in \mathbb{IR}^{n_x}$ be the initial guess interval such that $X^0 \subset Q$. Let $H(\mathbf{x}, U, Y)$ be an inclusion monotonic interval extension of* $\mathbf{h}$ *on $U \times Y$ for every $\mathbf{x} \in X^0$. The parametric Krawczyk method is defined as:*

$$\begin{aligned}
\mathbf{x}^k &:= m\left(X^k\right) \\
\mathbf{\Psi}^k &:= \left[m\left(J_{\mathbf{x}}(X^k, U, Y)\right)\right]^{-1} \\
K\left(\mathbf{x}^k, X^k, U, Y\right) &:= \mathbf{x}^k - \mathbf{\Psi}^k H(\mathbf{x}^k, U, Y) + \left(\mathbf{I} - \mathbf{\Psi}^k J_{\mathbf{x}}(X^k, U, Y)\right)(X^k - \mathbf{x}^k) \\
X^{k+1} &:= K(\mathbf{x}^k, X^k, U, Y) \cap X^k.
\end{aligned} \tag{10}$$

Similar to the Gauss-Seidel implementation for the parametric interval Newton method above, a componentwise sequential calculation of the K operator, with preconditioning, is preferred to give better convergence and enclosure properties. The implementation is defined in the following.

Definition 2.20 (Componentwise Parametric Krawczyk Method). *For $i = 1, 2, \ldots, n_x$:*

$$\begin{aligned}
K_i^k &:= x_i^k - b_i^k + \sum_{j=1}^{i-1} A_{i,j}(X_j^{k+1} - x_j^{k+1}) + \sum_{j=i}^{n_x} A_{i,j}(X_j^k - x_j^k) \\
X_i^{k+1} &:= K_i^k \cap X_i^k.
\end{aligned} \tag{11}$$

with $\mathbf{A}^k \in \mathbb{IR}^{n_x \times n_x}$ as $\mathbf{A}^k = \mathbf{I} - \mathbf{\Psi}^k J_{\mathbf{x}}(X^k, U, Y)$ and $\mathbf{b}^k \in \mathbb{R}^{n_x}$ as $\mathbf{b}^k = \mathbf{\Psi}^k H(\mathbf{x}^k, U, Y)$, where $\mathbf{\Psi}^k$ is taken as defined above.

In (Neumaier, 1990), it is shown that $N \subset K$ when implemented componentwise, for the non-parametric case. The extension to the parametric case can be made and the result holds. However, since N requires interval divisions, the parametric interval Newton method may not be recommended over the parametric Krawczyk method for all systems.

The last parametric interval method considered is the nested parametric interval successive substitution method. It is essentially a nested interval form of the generic successive substitution method for real numbers.

Definition 2.21 (Nested Parametric Interval Successive Substitution). *Let $\mathbf{f}(\cdot, \mathbf{u}, \mathbf{y}) : X \to X$ be an algebraic rearrangement of $\mathbf{h}(\cdot, \mathbf{u}, \mathbf{y}) : X \to \mathbb{R}^{n_x}$ such that $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{y})$. Let $F(X^k, U, Y)$ be an inclusion monotonic interval extension of* $\mathbf{f}$ *on $X^k \times U \times Y$ for every $X^k \subset X$. The Nested Parametric Interval Successive Substitution method is then defined as:*

$$\begin{aligned}
S\left(X^k, U, Y\right) &:= F\left(X^k, U, Y\right) \\
X^{k+1} &:= S\left(X^k, U, Y\right) \cap X^k.
\end{aligned} \tag{12}$$

The nested parametric interval successive substitution method offers a few novel properties. One property is that it does not require derivative information. Although this method is rather restrictive to relatively simple systems, it is a computationally inexpensive alternative. Also, under some assumptions, the converged interval will be the interval hull of the image. Thus it has the potential to yield the tightest possible enclosure of the range of the implicit function. The following example illustrates such a case and compares the three parametric interval methods.

Example 2.1. Let us set $\mathbf{q} = (\mathbf{u}, \mathbf{y})$. Let the steady-state model of interest be

$$\mathbf{h}(\mathbf{x}, \mathbf{q}) = \begin{pmatrix} x_1^2 + x_2^2 + q_1 x_1 + 4 \\ x_1 + q_2 x_2 \end{pmatrix} = \mathbf{0}$$

with $\mathbf{q} \in \Theta = [5, 7] \times [5, 7]$. Let $X^0 = [-1.5, 0] \times [0, 0.5]$. The successive substitution method can be applied if the problem is reformulated such that $\mathbf{h}(\mathbf{x}, \mathbf{q}) = \mathbf{0}$ is rewritten as $\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{q})$:

$$\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{q}) = \begin{pmatrix} -(x_1^2 + x_2^2 + 4)/q_1 \\ -x_1/q_2 \end{pmatrix}.$$

Likewise, the parametric interval Newton and parametric Krawczyk methods can also be applied. On this Θ interval, the exact and approximate ranges of the solution are

$$\begin{aligned} \hat{\mathbf{x}}^H(\Theta) &= \begin{pmatrix} \left[(5\sqrt{209} - 125)/52, (7\sqrt{1601} - 343)/100\right] \\ \left[(49 - \sqrt{1601})/100, (25 - \sqrt{209})/52\right] \end{pmatrix} \\ &\approx \begin{pmatrix} [-1.0137661\ldots, -0.629125\ldots] \\ [0.089875\ldots, 0.202753\ldots] \end{pmatrix}. \end{aligned}$$

The three algorithms were implemented using INTLAB (Rump, 1999). The algorithms terminate when $X^{k+1} = X^k$, which is an inherent stopping criterion for all interval iterations given outward rounding techniques are employed to approximate real intervals to machine precision. Tables I, II, and III in the Appendix display the results of the three interval algorithms. We see $X_S^*(\Theta) \subset X_N^*(\Theta) \subset X_K^*(\Theta)$, where $X_S^*(\Theta)$ approximates the interval hull to machine precision.

2.3. McCormick Relaxations

McCormick's relaxations (McCormick, 1976) are used in this algorithm to construct (nonsmooth) convex and concave relaxations of nonconvex functions, which are guaranteed to overestimate the nonconvex function. The direct application of McCormick relaxations is in calculating valid upper bounds for global maximization problems. In (Mitsos *et al.*, 2009), it was shown how to calculate McCormick relaxations and subgradients of (nonconvex or nonconcave) algorithms. They also automate the process using the `C++` library `libMC` (Chachuat, 2007), which can calculate relaxations and subgradients automatically, similar to automatic differentiation (AD). The proposed application and key result is in solving large nonconvex global optimization problems in a reduced-space (Mitsos *et al.*, 2009). Their results stop short of considering fixed-point algorithms such as Newton's method, where the number of iterations is not known *a priori*. We have extended these

McCormick-based relaxations of algorithms to include fixed-point algorithms. The theory behind the general construction of convex/concave relaxations of algorithms can be found in (Mitsos *et al.*, 2009), with the automatic implementation discussed in (Chachuat, 2007). This section will be used to summarize the basic concepts and define the nomenclature.

Definition 2.22 (Relaxation of Functions). *Given a convex set $Z \subset \mathbb{R}^n$ and a function $f : Z \to \mathbb{R}$, a convex function $f^c : Z \to \mathbb{R}$ is a convex relaxation (or convex underestimator) of f on Z if $f^c(\mathbf{z}) \leq f(\mathbf{z})$ for every $\mathbf{z} \in Z$. A concave function $f^C : Z \to \mathbb{R}$ is a concave relaxation (or concave overestimator) of f on Z if $f^C(\mathbf{z}) \geq f(\mathbf{z})$ for every $\mathbf{z} \in Z$.*

A key assumption made on $\mathbf{h}$ and g in this paper, is that they are factorable functions.

Definition 2.23 (Factorable Function). *A function is factorable if it is defined by a finite recursive composition of binary sums, binary products, and a given library of univariate intrinsic functions.*

Definition 2.24 (McCormick Relaxations). *The relaxations of a factorable function that are formed via the recursive application of rules for the relaxation of univariate composition, binary multiplication, and binary addition from convex and concave relaxations of the univariate intrinsic functions, without the introduction of auxiliary variables, are termed McCormick relaxations.*

In order to calculate valid convex/concave relaxations of implicit functions, it is necessary to first know a valid estimate of their range. Therefore, parametric interval Newton-type methods are required for constructing convex/concave relaxations of nonconvex/nonconcave implicit functions. The second step in constructing convex/concave relaxations of implicit functions is to rearrange the parametric system of equations, into a fixed-point form:

$$\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{y}).$$

In general, this can be done for any parametric system of equations $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{0}$ (Ortega and Rheinboldt, 1970). For example,

$$\boldsymbol{\Psi}\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{0} = \mathbf{x} - \mathbf{x} \Rightarrow \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mathbf{x} - \boldsymbol{\Psi}\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{y}),$$

again with $\boldsymbol{\Psi}$ as a preconditioning matrix. This is used to define the iteration:

$$\mathbf{x}^{k+1} := \mathbf{f}(\mathbf{x}^k, \mathbf{u}, \mathbf{y}).$$

Taking $\boldsymbol{\Psi} = \mathbf{J}_\mathbf{x}(\mathbf{x}, \mathbf{u}, \mathbf{y})^{-1}$ results in Newton's method, or taking $\boldsymbol{\Psi} = \mathbf{I}$ results in the Picard iteration. Now, convex/concave relaxations of $\mathbf{x}^{k+1}$ on $U \times Y$ can be calculated by applying a genralization of McCormick's method (Scott, Stuber, and Barton, 2009) to calculate convex/concave relaxations of $\mathbf{f}(\mathbf{x}^k, \mathbf{u}, \mathbf{y})$ on $U \times Y$ give convex/concave relaxations of $\mathbf{x}^k$ on $U \times Y$. Assuming the iteration $\mathbf{x}^{k+1} := \mathbf{f}(\mathbf{x}^k, \mathbf{u}, \mathbf{y})$ is a contraction mapping on X for each $(\mathbf{u}, \mathbf{y}) \in U \times Y$, the sequence of iterates will converge to the implicit function $\{\mathbf{x}^k(\mathbf{u}, \mathbf{y})\} \to \mathbf{x}(\mathbf{u}, \mathbf{y})$, for each $(\mathbf{u}, \mathbf{y}) \in U \times Y$. Therefore calculating convex/concave relaxations of $\mathbf{x}^{k+1}(\mathbf{u}, \mathbf{y})$ until $||\mathbf{x}^{k+1}(\mathbf{u}, \mathbf{y}) - \mathbf{x}^k(\mathbf{u}, \mathbf{y})|| \leq \epsilon$ results in valid convex and concave relaxations of the implicit function, $\mathbf{x}^c(\mathbf{u}, \mathbf{y})$ and $\mathbf{x}^C(\mathbf{u}, \mathbf{y})$, respectively, if the relaxations were initialized as rigorous lower and upper bounds respectively, on its range, calculated via a parametric interval Newton method. The general rules for constructing McCormick relaxations are outlined in (McCormick, 1976; Mitsos *et al.*, 2009; Scott, Stuber, and Barton, 2009).

2.4. Nonsmooth Programs

Since McCormick-based relaxations are nonsmooth, their incorporation into an NLP results in a nonsmooth convex NLP. Likewise, inclusion functions, which the proposed algorithm also relies on, are also potentially nonsmooth. Therefore, standard gradient-based methods for NLPs, such as *Sequential Quadratic Programming* (SQP) (Gill *et al.*, 2005), cannot be used. In (Mitsos *et al.*, 2009), the authors used the idea of *affine relaxations* of the nonsmooth convex relaxations to generate a new optimization problem with linear objective function and constraints whose solution is a lower-bound on the original nonconvex problem. In order to calculate affine relaxations, subgradient information is required.

Definition 2.25 (Subgradient). *Let $Z \subset \mathbb{R}^n$ be a nonempty convex set, $f^c : Z \to \mathbb{R}$ be convex and $f^C : Z \to \mathbb{R}$ be concave. A vector $\mathbf{s}^c \in \mathbb{R}^n$ is called a subgradient of f^c at $\bar{\mathbf{z}} \in Z$ if $f^c(\mathbf{z}) \geq f^c(\bar{\mathbf{z}}) + (\mathbf{s}^c)^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$ for all $\mathbf{z} \in Z$. Likewise, a vector $\mathbf{s}^C \in \mathbb{R}^n$ is called a subgradient of f^C at $\bar{\mathbf{z}} \in Z$ if $f^C(\mathbf{z}) \leq f^C(\bar{\mathbf{z}}) + (\mathbf{s}^C)^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$ for all $\mathbf{z} \in Z$.*

Definition 2.26 (Affine Relaxation). *Let $Z \subset \mathbb{R}^n$ be a nonempty convex set, $f^c : Z \to \mathbb{R}$ be a convex relaxation and $f^C : Z \to \mathbb{R}$ be a concave relaxation of $f : Z \to \mathbb{R}$ on Z, respectively. Let $\mathbf{s}^c$ and $\mathbf{s}^C$ be the corresponding subgradients of f^c and f^C at some point $\bar{\mathbf{z}} \in Z$, respectively. Then $f^{cl}(\mathbf{z}) \equiv f^c(\bar{\mathbf{z}}) + (\mathbf{s}^c)^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$ and $f^{Cl}(\mathbf{z}) \equiv f^C(\bar{\mathbf{z}}) + (\mathbf{s}^C)^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$ are an affine underestimator and affine overestimator of f on Z, respectively.*

One may also wish to solve the nonsmooth convex problem directly, rather than calculate a valid lower bound, using a nonsmooth optimization method such as a *bundle method*. However, this approach could be more expensive than the subgradient approach and the details for constrained problems need to be worked out. Solving nonsmooth optimization problems using bundle methods has been discussed in the literature (Mäkelä, 2001). These methods require that the objective function value and a subgradient can be evaluated at all desired points, a property that holds regardless of the embedded implicit functions from the discussion in Section 2.3. The rules for calculating subgradients are outlined in (Mitsos *et al.*, 2009). Likewise, in (Mitsos *et al.*, 2009), the authors show how subgradients can be propagated through McCormick relaxations. The calculation of subgradients of implicit functions through McCormick relaxations of implicit functions is analogous and can be automated using `libMC`. Therefore, subgradient information for the implicit functions, as well as the objective function, are available. This allows users to select the affine relaxation approach to calculate a valid upper bound on the solution of the nonsmooth maximization problem or to employ a novel nonsmooth solver directly.

3. Robust Simulation

Solving SIPs globally with explicit constraints using interval methods and McCormick's convex relaxations within the branch-and-bound framework is discussed in (Bhattacharjee *et al.*, 2005b). In (Bhattacharjee *et al.*, 2005b), the authors generate upper and lower bounding problems for the original SIP that are refined through the branch-and-bound framework. The sequences of upper and lower bounding solutions are guaranteed to converge to the true solution of the SIP under some relatively mild assumptions (Bhattacharjee *et al.*, 2005a; Bhattacharjee *et al.*, 2005b). In order to solve SIPs constrained by implicit functions, as in (3), a method similar to (Bhattacharjee *et al.*, 2005b) is applied.

3.1. Lower Bounding Problem

Solving the lower bounding problem (LBP) to local optimality is guaranteed to give a feasible point for the original SIP, if one exists (note that (3) is a maximization problem). That is, the set of feasible points of the LBP is a subset of the set of feasible points of the original SIP. Graphically, this means the feasible region of the LBP is an inner-approximation of the feasible region of the SIP. This result is especially useful in the case when simply guaranteeing feasibility of a proposed design is sufficient. The LBP is a finite nonsmooth reformulation of the original SIP known as an interval-constrained reformulation (ICR). It is only required to solve the ICR to local optimality, if a feasible point exists, for a valid lower bound on the solution. The ICR requires partitioning of the control domain U into n subintervals such that $U = \bigcup_{i=1}^{n} U^i$ and calculating an inclusion function, G, of the implicit semi-infinite constraint over each subset U^i. The inclusion functions are calculated as inclusion monotonic interval extensions of the constraint function g over U^i utilizing a parametric interval Newton-type method to generate valid inclusions of the implicit function $\mathbf{x}$ over U^i. For example, if the standard parametric interval Newton method is employed, its form would be:

$$\begin{aligned} N(\mathbf{x}^k, X^k, U, \mathbf{y}) &:= \mathbf{x}^k - J_{\mathbf{x}}(X^k, U, \mathbf{y})^{-1} H(\mathbf{x}^k, U, \mathbf{y}), \quad \mathbf{x}^k \in X^k \\ X^{k+1} &:= X^k \cap N(\mathbf{x}^k, X, U, \mathbf{y}). \end{aligned} \tag{13}$$

The algorithm (13) will converge to an interval-valued function $X(U, \mathbf{y})$. The calculation of G is then straightforward using the rules of interval arithmetic. It is explained in (Bhattacharjee *et al.*, 2005a) that given the existence of a Slater point arbitrarily close to a maximizer, as the width of the subintervals approaches degeneracy, the solution value of the ICR approaches the true global solution value of the original SIP from below, for this formulation. The ICR is defined in the following.

Definition 3.1 (Interval-Constrained Reformulation). *Let $G^L(X(U^i, \mathbf{y}), U^i, \mathbf{y})$ be the lower bound of an inclusion monotonic interval extension of $g(\mathbf{x}(\mathbf{u}, \mathbf{y}), \mathbf{u}, \mathbf{y})$ on the i^{th} subinterval U^i. The lower-bounding ICR is as follows:*

$$\begin{aligned} \eta^{LBD} &= \max_{\mathbf{y} \in Y, \eta \in \mathbb{R}} \eta \\ \text{s.t. } & \eta \leq G^L(X(U^i, \mathbf{y}), U^i, \mathbf{y}), \quad \forall i = 1, \ldots, n. \end{aligned} \tag{14}$$

Solving (14) to local optimality yields a feasible point, if one exists, and is a lower bound on the solution value of the SIP, $\eta^{LBD} \leq \eta^*$. It should be noted, however, that the implicit function G^L is nonsmooth, thus (14) is a nonsmooth NLP.

3.2. Upper Bounding Problem

The upper bounding problem (UBP) is based on the discretization technique where the semi-infinite constraint is replaced by a finite collection of constraints evaluated at each point in a finite subset of the control set U. The discretized program is as follows:

$$\begin{aligned} &\max_{\mathbf{y}\in Y,\eta\in\mathbb{R}} \eta \\ &\text{s.t. } \eta \leq g(\mathbf{x}(\mathbf{u}_i,\mathbf{y}),\mathbf{u}_i,\mathbf{y}), \ \ \forall i = 1,\dots,n, \end{aligned} \tag{15}$$

where $\mathbf{u}_i \in U$ denotes the i^{th} realization of the control variable $\mathbf{u}$. As explained in (Bhattacharjee *et al.*, 2005b), as the number of discretization points, n, increases, the solution of this finite relaxation approaches the solution to the SIP from above, for this formulation. However, since the feasible set of (15) is likely nonconvex, in order to guarantee its solution yields a valid upper bound on the original SIP, it must be solved to global optimality. Assuming (15) is composed of factorable functions, the results of McCormick convex/concave relaxations of fixed-point iterations on Y can be utilized to generate a valid (nonsmooth) overestimating problem, with a convex feasible set, as follows:

$$\begin{aligned} \eta^{UBD} = &\max_{\mathbf{y}\in Y,\eta\in\mathbb{R}} \eta \\ &\text{s.t. } \eta \leq g^C(\mathbf{x}^c(\mathbf{u}_i,\mathbf{y}),\mathbf{x}^C(\mathbf{u}_i,\mathbf{y}),\mathbf{u}_i,\mathbf{y}), \ \forall i = 1,\dots,n, \end{aligned} \tag{16}$$

again where the superscripts c and C denote the convex and concave relaxations, respectively. The feasible set of (16) is a convex outer approximation of the feasible set of the SIP. Solving (16) yields a valid upper bound on the solution to the SIP, $\eta^{UBD} \geq \eta^*$.

3.3. Branch & Bound

The Branch & Bound (B&B) algorithm for solving nonconvex optimization problems globally is discussed in (Horst and Tuy, 1996). The extension to SIPs is described in (Bhattacharjee *et al.*, 2005b). The idea here is the same for SIPs with an implicit semi-infinite constraint. The B&B framework relies on the solution of LBP and UBP over a finite number of subsets, or nodes, of the decision domain, in this case Y. If it is known that a node cannot contain a global maximizer $\mathbf{y}^*$, the node is said to be *fathomed*. The general rule for fathoming nodes is to compare the values of the η^{UBD} and η^{LBD} of each node. For instance, if $\eta_k^{LBD} > \eta_j^{UBD}$, a global maximizer cannot be in node j, and node j is said to be fathomed. Once a node is fathomed, it is no longer considered in the search space. This procedure continues by refining each node that has not been fathomed. The B&B algorithm, as applied here, is finitely-convergent to ϵ-optimality ($\eta^{UBD} - \eta^{LBD} \leq \epsilon_{tol}, \ \ \epsilon_{tol} > 0$).

3.4. Solution Algorithm

1. Initialize algorithm.
   a) Initialize stack of nodes $\Sigma = \{Y\}$.
   b) Set convergence tolerance ϵ_{tol}, $LBD = -\infty$, $UBD = +\infty$.
   c) Define partitioning and discretization sequences for U for the UBP and LBP.
2. Check if $\Sigma = \emptyset$.
   a) If true, terminate.
   b) Else select and delete a node Y^k from Σ according to a node selection rule/heuristic.
3. LBP.
   a) Generate valid inclusion function $G(X(U^i, \mathbf{y}), U^i, \mathbf{y})$ over each subinterval U^i using a parametric interval Newton-type method.
   b) Solve ICR (14) on Y^k for feasible point (if one exists), get η_k^{LBD}.
   c) If $\eta_k^{LBD} > LBD$, set $LBD := \eta_k^{LBD}$.
4. UBP.
   a) Calculate valid bounds for implicit semi-infinite constraint on Y^k at each discrete $\mathbf{u}_i \in U$ using a parametric interval Newton-type method.
   b) Generate concave relaxation on Y^k and solve UBP (16) on Y^k, get η_k^{UBD}.
   c) Set $UBD := \max_{Y_k \in \Sigma} \eta_k^{UBD}$.
5. Check $\eta_k^{UBD} < LBD$ or $\eta_k^{UBD} = -\infty$.
   a) If true, go to 2 (Y^k has been fathomed).
6. Check $UBD - LBD \leq \epsilon_{tol}$.
   a) If true, set $\eta^* = LBD$, terminate algorithm.
7. Branching
   a) Branch on Y^k according to a branching rule.
   b) Push new nodes onto stack Σ.
   c) Go to 2.

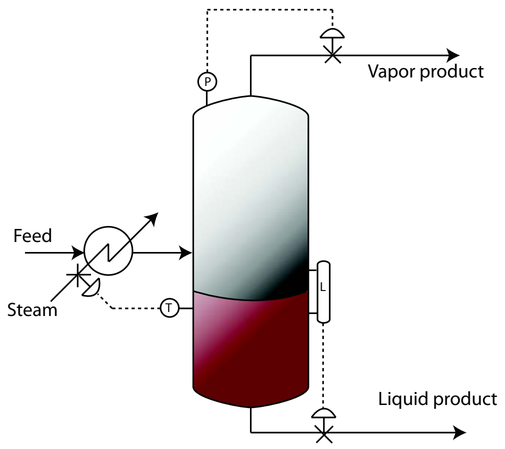

Figure 1. The continuous equilibrium flash separator as taken from (King, 1980).

3.5. Numerical Example

Example 3.1 (Flash Separation of Benzene and Toluene). As an illustrative example, take the benzene/toluene flash separation originally introduced in (King, 1980), shown in Figure 1. We wish to guarantee the process meets robustly, a preset performance specification on the separation. Uncertainty in the separator temperature τ, that could be caused by fluctuations in the incoming stream temperature or faulty sensor readings, will be taken into account. The pressure in the separator, p, will be the only thing we can control, say via the valve on the vapor line. The uncertainty temperature interval will be $T = [80, 110]$ °C and the control interval will be $P = [90, 100]$ torr. The performance constraint is such that the cut fraction α must be less-than or equal to 0.7 ($g(\alpha, p, \tau) = \alpha - 0.7$). Of course, it should be noted that any $\alpha \notin [0, 1]$ is non-physical. This problem is stated as the following robust simulation problem:

$$\begin{aligned} &\max_{\tau\in T,\eta\in\mathbb{R}} \eta \\ \text{s.t. } &\eta \leq \min_{\alpha\in A, p\in P} g(\alpha, p, \tau) \\ &\quad \text{s.t. } h(\alpha, p, \tau) = 0 \\ &P = [90, 100] \\ &T = [80, 110] \\ &A = [0, 1]. \end{aligned} \tag{17}$$

Solving the steady-state process model, h, for the cut-fraction as an implicit function of pressure and temperature, $\alpha(p, \tau)$, the equivalent SIP can be formulated as follows:

$$\begin{aligned} &\max_{\tau\in T,\eta\in\mathbb{R}} \eta \\ &\text{s.t. } \eta \leq g(\alpha(p,\tau),p,\tau) = \alpha(p,\tau) - 0.7, \quad \forall p \in P \\ &P = [90,100] \\ &T = [80,110]. \end{aligned} \tag{18}$$

The steady-state process model equation is written as:

$$h(\alpha,p,\tau) = \sum_{i=1}^{2} \frac{z_i(K_i(\tau,p)-1)}{(K_i(\tau,p)-1)\alpha+1} = 0, \tag{19}$$

where z_i is the mole-fraction of component i in the feed, taken to be 0.5, and $K_i : T \times P \to \mathbb{R}$ is the vapor-liquid equilibrium distribution coefficient of component i, defined as

$$K_i(\tau,p) = \frac{p_i^{sat}(\tau)}{p} \tag{20}$$

with

$$\log_{10}[p_i^{sat}(\tau)] = A_i - \frac{B_i}{C_i+\tau}. \tag{21}$$

The Antoine constants A_i, B_i, C_i are given in the Appendix in Table IV. The pressure-domain is partitioned into two subintervals at the root-node of the B&B tree, $P_1 = [90,95]$ and $P_2 = [95,100]$, for use in the LBP. Likewise, the P-domain is discretized into 3 points corresponding to the endpoints of the subintervals P_1 and P_2, $p_1 = 90$, $p_2 = 95$, and $p_3 = 100$. The parametric interval Newton method (7) was applied to calculate valid ranges for the implicit function at each discrete pressure value and to calculate the inclusion functions valid over each pressure subinterval. Corresponding concave relaxations of the implicit semi-infinite constraint on T were calculated automatically with `libMC` using a convergence tolerance on Newton's method of 10^{-8}. Figure 2 is a plot of the discrete implicit constraints, g, and their corresponding concave relaxations. Figure 3 shows the plots of the discrete implicit constraints g and the corresponding inclusion lower bound G^L. For this simple example, the SIP can be solved at the root node without branching. From the figures, it is clear that as $\tau \to 110$ we have $g(\alpha(p,\tau),p,\tau)$ moving positive. Thus $\eta^* > 0$, implying that for all realizations of uncertainty in the temperature within the interval T, there does not exist a pressure setting such that $\alpha \leq 0.7$ for all temperatures. However, if we change the performance constraint to $\alpha \leq 0.9$, we find $\eta^* < 0$, implying robust feasibility of the process.

4. Conclusion

A model-based approach to designing process systems is required to guarantee the system will meet all performance specifications robustly, given various uncertainties in the process. The high level of complexity of such simulations warrants the need for a reduced-space approach. Solving steady-state model equations for process state variables as functions of the controls and uncertainty parameters, offers a potentially large reduction in size, making robust simulation and design under uncertainty

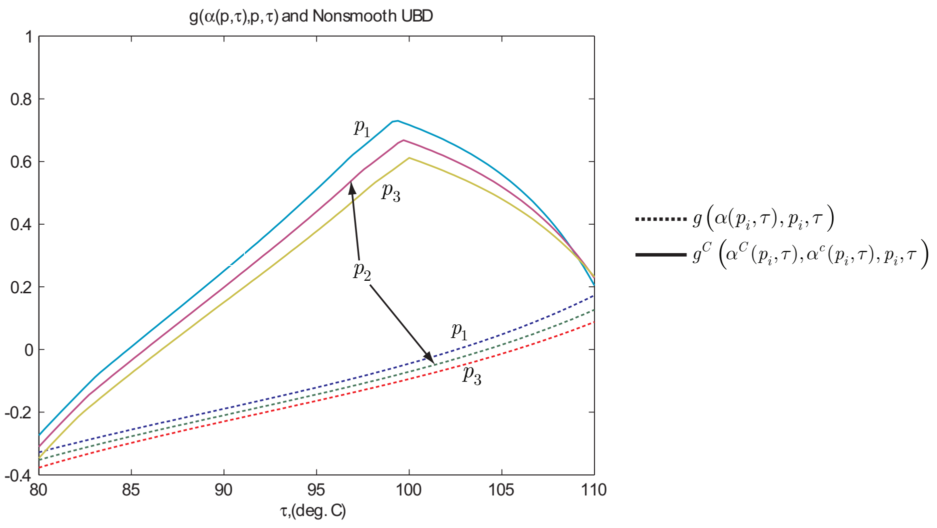

Figure 2. The implicit constraints and their corresponding concave relaxations.

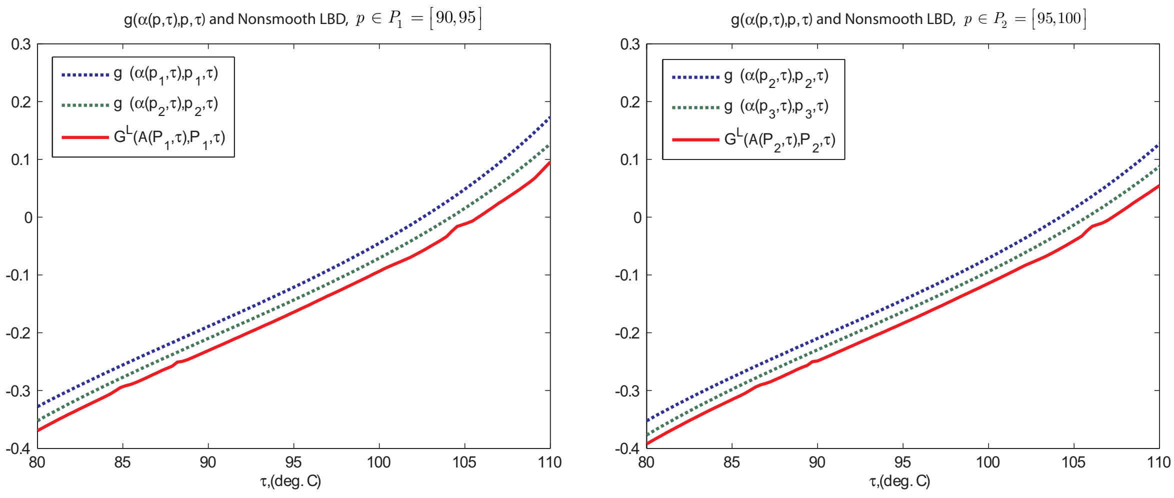

Figure 3. The implicit constraints and their corresponding inclusion lower bound.

more tractable. However, this technique results in a semi-infinite optimization problem in which the semi-infinite constraint is implicitly defined. The authors have shown how, with the application of the novel ideas of parametric interval Newton-type methods and McCormick-based relaxations of algorithms, the implicit SIP can be solved to ϵ-global optimality. The proposed rigorous method makes use of a lower bounding problem whose feasible region is a nonsmooth, nonconvex inner-approximation to the SIP feasible region, and an upper bounding problem whose feasible region is a nonsmooth convex outer-approximation of the SIP feasible region. The approximations are continually refined while branching in the decision domain. The process in question is said to be robust if the SIP solution value $\eta^* \leq 0$.

Appendix

Table I. After 44 iterations (0.13sec), the parametric interval successive substitution method terminates. Note the interval encloses the hull exactly.

Parametric Interval Successive Substitution			
i	x_i^{*L}	x_i^{*U}	$w(E(X_i^*))$
1	-1.013766...	-0.629125...	0
2	0.0898750...	0.202753...	0

Table II. After 24 iterations (0.24sec), the parametric interval Newton method with interval Gauss-Seidel terminates.

Parametric Interval Newton's Method			
i	x_i^{*L}	x_i^{*U}	$w(E(X_i^*))$
1	-1.04243...	-0.49276...	0.16503...
2	0.047379...	0.208486...	0.04823...

Table III. After 54 iterations (0.57sec), the parametric Krawczyk method implemented componentwise terminates.

Parametric Krawczyk Method			
i	x_i^{*L}	x_i^{*U}	$w(E(X_i^*))$
1	-1.04243...	-0.49276...	0.16503...
2	0.047379...	0.208486...	0.04823...

Table IV. Antoine constants for benzene and toluene (Elliot and Lira, 1999).

i	A_i	B_i	C_i
1: Tol.	6.95087	1342.31	219.187
2: Benz.	6.87987	1936.01	258.451

Acknowledgements

The authors would like to acknowledge the Chevron University Partnership Program for supporting this work through the MIT Energy Initiative (UPP/MITEI).

References

Bertsekas, Dimitri P. Nonlinear Programming. Athena Scientific, Belmont, MA, Second Ed., 2003.

Bhattacharjee, B., Green Jr. W. H., and P. I. Barton. Interval Methods for Semi-Infinite Programs *Computational Optimization and Applications*, 30:63–93, 2005.

Bhattacharjee, B., Lemonidis, P., Green Jr. W. H., and P. I. Barton. Global Solution of Semi-Infinite Programs. *Math. Program.*, Ser. B 103:283–307, 2005.

Chachuat, Benoit. libMC: A Numeric Library for McCormick Relaxation of Factorable Functions. `http://yoric.mit.edu/libMC/`, 2007.

Elliot, J. Richard, and Carl T. Lira. Introductory Chemical Engineering Thermodynamics. Prentice-Hall, New Jersey, 1999.

Gill, P. E., Murray, W. and M. A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review.*, 47(1):99–131, 2005.

Haleman, K. P., and Grossman, I. E. Optimal Process Design Under Uncertainty. *AIChE Journal*, 29(3):425–433, 1983.

Hansen, Eldon and G. William Walster. Global Optimization Using Interval Analysis. Marcel Dekker, New York, Second Ed., 2004.

Horst, R., and H. Tuy. Global Optimization: Deterministic Approaches. Springer-Verlag, New York, 1997.

Kearfott, R. Baker. Preconditioners for the Interval Gauss-Seidel Method. *SIAM J. Num. An.*, 27(3):804–822, 1990.

Kearfott, R. Baker. Rigorous Global Search: Continuous Problems. Kluwer Academic Publishers, Boston, 1996.

King, J. C. Separation Processes. McGraw-Hill, New York, Second Ed., 1980.

Mäkelä, Marko M. Survey of Bundle Methods for Nonsmooth Optimization. *Optimization Methods and Software*, 17(1):1–29, 2001.

McCormick, Garth P. Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimating Problems. *Math. Prog.*, 10:147-175, 1976.

Mitsos, A., Chachuat, B., and P. I. Barton. McCormick-Based Relaxations of Algorithms. *SIAM J. Opt.*, 20(2), 573-601, 2009.

Moore, Ramon E. Methods and Applications of Interval Analysis. SIAM, Philadelphia, 1979.

Neumaier, Arnold. Interval Methods for Systems of Equations. Cambridge University Press, Cambridge, 1990.

Ortega, J. M. and W. C. Rheinboldt. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press Inc., Boston, 1970.

Rump, S. M. INTLAB-INTerval LABoratory. in *Developments in Reliable Computing*, Tibor Csendes, ed., Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104.

Scott, J. K., Stuber, M. D., and P. I. Barton. Generalized McCormick Relaxations. *submitted to Journal on Global Optimization*, 2009.