

An LP Strategy for the Interval-Newton Method in Deterministic Global Optimization

Youdong Lin and Mark A. Stadtherr*

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

August 2003

(revised, November 2003)

*Author to whom all correspondence should be addressed. Phone: (574) 631-9318; Fax: (574) 631-8366;
E-mail: markst@nd.edu

Abstract

A strategy is described for using linear programming (LP) to bound the solution set of the linear interval equation system that must be solved in the context of the interval-Newton method for deterministic global optimization. An implementation of this technique is described in detail, and several important issues are considered. These include selection of the interval corner required by the LP strategy, and determination of rigorous bounds on the solutions of the LP problems. The impact of using a local minimizer for updating the upper bound on the global minimum in this context is also considered. The procedure based on these techniques, LISS_LP, is demonstrated using several global optimization problems, with focus on problems arising in chemical engineering. Problems with a very large number of local optima can be effectively solved, as well as problems with a relatively large number of variables.

Keywords: Global Optimization, Linear Programming, Interval-Newton, Parameter Estimation, Numerical Method

1 Introduction

Many optimization problems encountered in chemical engineering applications are nonlinear and nonconvex, and thus potentially have multiple local optima. Typically in such problems it is desired to seek the *global* optimum, though in some problems it might be of interest to find *all* the local optima, or even all stationary points. One approach that can be used to address these problems deterministically is the use of interval analysis. In recent years, it has been shown that strategies based on an interval-Newton approach can be used to reliably solve a wide variety of global optimization and nonlinear equation solving problems in chemical engineering, including computation of fluid phase equilibrium from activity coefficient models,¹⁻³ cubic equation-of state (EOS) models⁴⁻⁶ and statistical associating fluid theory,⁷ calculation of critical points from cubic EOS models,⁸ location of azeotropes⁹ and reactive azeotropes,¹⁰ computation of solid-fluid equilibrium,^{11,12} parameter estimation using standard least squares¹³ and error-in-variables (EIV),^{14,15} and calculation of adsorption in nanoscale pores from a density function theory model.¹⁶ In each case, the interval approach provides a mathematical and computational guarantee either that all solutions have been located in a nonlinear equation solving problem or that the global optimum has been found in an optimization problem. Another powerful approach for deterministic global optimization, which has been used in several applications, is the use of convex underestimating functions in connection with a branch-and-bound strategy, as implemented, for example, in the α -BB approach.^{17,18}

A drawback to the interval approach, as well as to other approaches for deterministic global optimization, is the potentially high computational cost. One way to improve the efficiency of the interval-Newton approach is to more tightly bound the solution set of the linear interval equation

system that is at the core of this approach. Recently, Lin and Stadtherr¹⁹ proposed a new bounding strategy based on the use of linear programming (LP) techniques, and also gave some examples demonstrating its potential. Using this approach it is possible to exactly (within round out), determine the desired bounds on the solution set of the linear interval system. By providing tight interval bounds on the solution set, the goal is to more quickly contract intervals that may contain solutions of the nonlinear system, as well as to more quickly identify intervals that contain a unique solution, and intervals that contain no solution, thus leading to an overall improvement in computational efficiency.

In this paper, it is explained in detail how the basic ideas presented earlier¹⁹ can be put into practice. Several important implementation issues will be considered: 1. It is shown how the LP strategy can be combined with a simple, low-overhead pivoting preconditioning technique^{20–22} and a real-point selection scheme²² to improve performance. 2. The LP strategy requires the selection of an interval corner; a heuristic procedure is given for determining which corner should be selected. 3. In order to ensure the reliability of the result, a procedure for determining rigorous error bounds on the solution of the LP problems is given. The procedure based on these techniques, LISS_LP, will then be further demonstrated using several global optimization problems, with focus on problems arising in chemical engineering.

In the next section, as background, we provide a brief outline of the interval-Newton approach for nonlinear equation solving and global optimization. In Section 3, the basic ideas of the LP-based strategy for bounding the solution set of the interval-Newton equation are summarized. Then, in Section 4, we explain in detail how to implement this strategy, addressing the implementation issues noted above. Finally, in Section 5, several examples are used to demonstrate the methodology and evaluate its performance.

2 Background

Several good introductions to interval computations are available.^{23–26} Of particular interest here is the interval-Newton method. Given an $n \times n$ nonlinear equation system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ with a finite number of real roots in some initial interval, this technique provides the capability to find tight enclosures of all the roots of the system that lie within the given initial interval. For the unconstrained minimization of $\phi(\mathbf{x})$, a common approach is to seek stationary points, that is, to solve the nonlinear system $\mathbf{f}(\mathbf{x}) = \nabla\phi(\mathbf{x}) = \mathbf{0}$. The global optimum will be one of roots of this nonlinear equation system, but there may be other roots as well, representing local optima and saddle points. To identify the global optimum, it is critical that none of the roots be missed, and such a guarantee can be provided by the interval-Newton approach. For a constrained optimization problem, the interval-Newton method can be applied to solve the KKT or Fritz-John conditions. An outline of the interval-Newton methodology is given here. More details are available elsewhere.^{24,25,27}

Given some initial interval $\mathbf{X}^{(0)}$, the interval-Newton algorithm is applied to a sequence of subintervals. For a subinterval $\mathbf{X}^{(k)}$ in the sequence, the first step is the *function range test*. An *interval extension* $\mathbf{F}(\mathbf{X}^{(k)})$ of the function $\mathbf{f}(\mathbf{x})$ is calculated. An interval extension provides upper and lower bounds on the range of values that a function may have in a given interval. It is often computed by substituting the given interval into the function and then evaluating the function using interval arithmetic. Thus, the interval extension is often wider than the actual range of function values, but it always includes the actual range. If there is any component of the interval extension $\mathbf{F}(\mathbf{X}^{(k)})$ that does not include zero, then the interval can be discarded, since no solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ can exist in this interval. The next subinterval in the sequence may then be considered. Otherwise, testing of $\mathbf{X}^{(k)}$ continues. During this step, other interval-based techniques

(e.g., constraint propagation) may also be applied to try to shrink $\mathbf{X}^{(k)}$ before proceeding.

For a global minimization problem, the next step is the *objective range test*. The interval extension $\Phi(\mathbf{X}^{(k)})$, containing the range of $\phi(\mathbf{x})$ over $\mathbf{X}^{(k)}$ is computed. If the lower bound of $\Phi(\mathbf{X}^{(k)})$ is greater than a known upper bound on the global minimum, then $\mathbf{X}^{(k)}$ can be discarded since it cannot contain the global minimum and need not be further tested. If it is known that $\mathbf{X}^{(k)}$ contains a point that can be used to update (reduce) the upper bound on the global minimum (i.e., if the upper bound of $\Phi(\mathbf{X}^{(k)})$ is less than the current upper bound on the global minimum), then this update is performed. This can be done in many different ways. A simple, cheap approach that we have used effectively is to evaluate $\phi(\mathbf{x})$ at the midpoint of $\mathbf{X}^{(k)}$ and use this to update the upper bound. Another approach is to use a local minimization routine starting at the midpoint of $\mathbf{X}^{(k)}$. For this purpose, we use the simple, low-overhead direct search algorithm of Hooke and Jeeves.^{28,29} Use of the local minimizer involves additional computational overhead, but should lead to a better upper bound on the global minimum. Whether this is a profitable tradeoff is explored below in Section 5. In cases when all the stationary points are desired rather than just the global minimum, this test step can be turned off.

The next step is the *interval-Newton test*. The linear interval equation system

$$\mathbf{F}'(\mathbf{X}^{(k)})(\mathbf{N}^{(k)} - \mathbf{x}^{(k)}) = -\mathbf{f}(\mathbf{x}^{(k)}), \quad (1)$$

is solved for a new interval $\mathbf{N}^{(k)}$, where $\mathbf{F}'(\mathbf{X}^{(k)})$ is an interval extension of the Jacobian of $\mathbf{f}(\mathbf{x})$, and $\mathbf{x}^{(k)}$ is an arbitrary point in $\mathbf{X}^{(k)}$. It has been shown^{24–26} that any root contained in $\mathbf{X}^{(k)}$ is also contained in the *image* $\mathbf{N}^{(k)}$. This implies that if the intersection between $\mathbf{X}^{(k)}$ and $\mathbf{N}^{(k)}$ is empty, then no root exists in $\mathbf{X}^{(k)}$, and also suggests the iteration scheme $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} \cap \mathbf{N}^{(k)}$. In addition, it has also been shown^{24–26} that, if $\mathbf{N}^{(k)} \subset \mathbf{X}^{(k)}$, then there is a *unique* root contained in

$\mathbf{X}^{(k)}$ and thus in $\mathbf{N}^{(k)}$. Thus, after computation of $\mathbf{N}^{(k)}$ from Eq. (1), there are three possibilities:

- (1) $\mathbf{X}^{(k)} \cap \mathbf{N}^{(k)} = \emptyset$, meaning there is no root in the current interval $\mathbf{X}^{(k)}$ and it can be discarded;
- (2) $\mathbf{N}^{(k)} \subset \mathbf{X}^{(k)}$, meaning that there is *exactly* one root in the current interval $\mathbf{X}^{(k)}$;
- (3) neither of the above, meaning that no conclusion can be drawn. In the last case, if $\mathbf{X}^{(k)} \cap \mathbf{N}^{(k)}$ is sufficiently smaller than $\mathbf{X}^{(k)}$, then the interval-Newton test can be reapplied to the resulting intersection. Otherwise, the intersection is bisected, and the resulting two subintervals are added to the sequence (stack) of subintervals to be tested. If an interval containing a unique root has been identified, then this root can be tightly enclosed by continuing the interval-Newton iteration, which will converge quadratically to a desired tolerance (on the enclosure diameter).

This approach is referred to as an interval-Newton/generalized-bisection (IN/GB) method. At termination, when the subintervals in the sequence have all been tested, either all the real roots of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ have been tightly enclosed or it is determined that no root exists. Applied to nonlinear equation solving problems, this can be regarded as a type of branch-and-prune scheme on a binary tree. Applied to global optimization problems, the use of the objective range test makes this a type of branch-and-bound scheme on a binary tree.

Clearly, the solution of the linear interval system given by Eq. (1) is essential to this approach. To see the issues involved in solving such a system, consider the general linear interval system $\mathbf{A}\mathbf{z} = \mathbf{B}$, where the matrix \mathbf{A} and the right-hand-side vector \mathbf{B} are interval-valued. The solution set S of this system is defined by $S = \left\{ \mathbf{z} \mid \tilde{\mathbf{A}}\mathbf{z} = \mathbf{b}, \tilde{\mathbf{A}} \in \mathbf{A}, \mathbf{b} \in \mathbf{B} \right\}$. However, in general this set is not an interval and may have a very complex, polygonal geometry. Thus to “solve” the linear interval system, one instead seeks an interval \mathbf{Z} containing S . Computing the interval hull (the tightest interval containing S) is NP-hard,³⁰ but there are several methods for determining an interval \mathbf{Z} that contains but overestimates S . Various interval-Newton methods differ in how they

solve Eq. (1) for $\mathbf{N}^{(k)}$ and thus in the tightness with which the solution set is enclosed. By obtaining bounds that are as tight as possible, the overall performance of the interval-Newton approach can be improved, since with a smaller $\mathbf{N}^{(k)}$ the volume of $\mathbf{X}^{(k)} \cap \mathbf{N}^{(k)}$ is reduced, and it is also more likely that either $\mathbf{X}^{(k)} \cap \mathbf{N}^{(k)} = \emptyset$ or $\mathbf{N}^{(k)} \subset \mathbf{X}^{(k)}$ will be satisfied. Thus, intervals that may contain solutions of the nonlinear system are more quickly contracted, and intervals that contain no solution or that contain a unique solution may be more quickly identified, all of which leads to a likely reduction in the number of bisections needed.

Frequently, $\mathbf{N}^{(k)}$ is computed component-wise using an interval Gauss-Seidel approach, preconditioned with an inverse-midpoint matrix. Though the inverse-midpoint preconditioner is a good general-purpose preconditioner, it is not always the most effective approach.²⁴ Recently, a hybrid preconditioning approach, which combines a simple pivoting preconditioner with the standard inverse-midpoint scheme, has been described by Gau and Stadtherr²² and shown to achieve substantially more efficient computational performance than the inverse-midpoint preconditioner alone. However, it still cannot yield the tightest enclosure of the solution set, which, as noted above, is in general an NP-hard problem. Lin and Stadtherr¹⁹ have recently suggested an LP-based strategy for solving the linear interval system, Eq. (1), arising in the context of interval-Newton methods. Using this approach, exact component-wise bounds on the solution set can be calculated, while avoiding exponential time complexity. The basic idea behind this strategy is outlined in the next section, and then in Section 4 details of its implementation are given.

3 An LP Strategy for the Interval-Newton Method

Many types of methods have been proposed for bounding the solution set of a system of linear

interval equations. One such method is based on the use of LP techniques.^{23,31–33}

Consider the linear interval system $Az = B$. Oettli & Prager³⁴ showed that the solution set S is determined by the constraints:

$$\left| \hat{A}z - \hat{B} \right| \leq \Delta A |z| + \Delta B, \quad (2)$$

where \hat{A} is the component-wise midpoint matrix of the interval matrix A , ΔA is the component-wise half-width (radius) matrix of A , and similarly \hat{B} and ΔB are the midpoint and radius of B . Eq. (2) is not directly useful for computing bounds on the solution set because of the absolute value operation on the right-hand side. In general, the solution may lie in all 2^n orthants for an n -dimensional problem. In each orthant, each component of z keeps a constant sign, and thus the absolute value can be dropped. For a given orthant, define the diagonal matrix D_α by

$$(D_\alpha)_{jj} = \begin{cases} 1 & z_j \geq 0 \\ -1 & z_j \leq 0 \end{cases} \quad j = 1, 2, \dots, n. \quad (3)$$

Thus $|z| = D_\alpha z$ and $z = D_\alpha |z|$. Eq. (2) becomes:

$$\left| \hat{A}z - \hat{B} \right| \leq \Delta A D_\alpha z + \Delta B. \quad (4)$$

This can be rearranged to the set of linear inequalities

$$\begin{pmatrix} \hat{A} - \Delta A D_\alpha \\ -\hat{A} - \Delta A D_\alpha \end{pmatrix} z \leq \begin{pmatrix} \overline{B} \\ -\underline{B} \end{pmatrix}, \quad (5)$$

where the underline and overline denote lower and upper interval bounds, respectively. To determine the tightest interval enclosing the solution set, one can then solve, in each orthant, the set of $2n$

optimization problems

$$\max_{\mathbf{z}} z_j, \quad j = 1, 2, \dots, n, \quad (6)$$

$$\min_{\mathbf{z}} z_j, \quad j = 1, 2, \dots, n, \quad (7)$$

each with the $2n$ linear inequality constraints given by Eq. (5). These can be solved using linear programming (LP) techniques. However, in general, there are 2^n orthants and so the solution time complexity will be exponential, as expected since this problem is known to be NP-hard.

In the context of the interval-Newton method, however, the exponential time complexity can be avoided. This is because only that part of the solution set of Eq. (1) that intersects $\mathbf{X}^{(k)}$ needs to be found. Consider the choice of the real point $\mathbf{x}^{(k)}$ in Eq. (1). Here $\mathbf{x}^{(k)}$ is an arbitrary point in $\mathbf{X}^{(k)}$ typically taken to be the midpoint. However, if $\mathbf{x}^{(k)}$ is chosen to be a corner of $\mathbf{X}^{(k)}$ instead, the part of the solution set for $\mathbf{N}^{(k)} - \mathbf{x}^{(k)}$ of Eq. (1) that intersects $\mathbf{X}^{(k)}$ lies in just one orthant. Thus, in the context of interval-Newton, only $2n$ LP subproblems, each with $2n$ constraints, needs to be solved. Furthermore, the LP subproblems have properties that can be exploited. First, all the $2n$ subproblems share the same constraints; that is, they all have the same feasible region. Thus, an initial feasible basis for the LP subproblems needs to be found only once. Second, the objective function of each subproblem consists of just one variable. This makes the problem much simpler since it is not necessary, as it is in the general case, to calculate the gain in objective value when choosing variables to enter and exit the basis.

LISS_LP (Linear Interval System Solver by Linear Programming) is a procedure that we have developed based on the above scheme. This procedure is used to replace the inverse-midpoint preconditioned Gauss-Seidel method for solving Eq. (1).

4 Implementation of LISS_LP

As noted above, the interval-Newton procedure is generally implemented by using an interval Gauss-Seidel method to bound the solution to Eq. (1). In order to obtain reasonably good bounds when this solution method is used, it is necessary to use a preconditioner, which is generally taken to be the inverse midpoint matrix. When interval-Newton is implemented using LISS_LP, Eq. (1) is solved using an LP approach that guarantees tight bounds without requiring a preconditioner. Nevertheless, the pivoting preconditioner scheme described by Gau and Stadtherr²² can still be used as a simple, low overhead means to try to shrink, or perhaps eliminate, $\mathbf{X}^{(k)}$. Thus, we incorporate this procedure in LISS_LP. Gau and Stadtherr²² also described another technique, based on adjusting the value of the real point $\mathbf{x}^{(k)}$, to further shrink, or perhaps eliminate, $\mathbf{X}^{(k)}$. This procedure is also incorporated in LISS_LP, where it is also used to help determine the corner of $\mathbf{X}^{(k)}$ to be used as $\mathbf{x}^{(k)}$ in the LP strategy. This is important since the selection of the real point $\mathbf{x}^{(k)}$ can have a significant impact on the size of the solution set of Eq. (1).

4.1 Pivoting Preconditioner and Selection of Real Point

If the interval-Newton equation is first preconditioned using a real matrix $Y^{(k)}$, then the preconditioned linear interval equation system can be expressed as

$$Y^{(k)}F'(\mathbf{X}^{(k)})(\mathbf{N}^{(k)} - \mathbf{x}^{(k)}) = -Y^{(k)}\mathbf{f}(\mathbf{x}^{(k)}). \quad (8)$$

Defining \mathbf{y}_i as the i -th row of the preconditioning matrix $Y^{(k)}$, \mathbf{A}_i as the i -th column of the interval Jacobian $F'(\mathbf{X}^{(k)})$, and letting $\mathbf{X} = \mathbf{X}^{(k)}$ and $\mathbf{x} = \mathbf{x}^{(k)}$, then Eq. (8) can be written as

$$N_i = x_i - \frac{\mathbf{y}_i\mathbf{f}(\mathbf{x}) + \sum_{j=1, j \neq i}^n \mathbf{y}_i\mathbf{A}_j(X_j - x_j)}{\mathbf{y}_i\mathbf{A}_i}, \quad i = 1, \dots, n, \quad (9)$$

where $\mathbf{y}_i \mathbf{A}_j$ indicates the inner product of the real row vector \mathbf{y}_i and the interval column vector \mathbf{A}_j . In a pivoting preconditioner,²¹ only one element of the preconditioning row \mathbf{y}_i , called the *pivot* element, is nonzero, and it is assigned a value of one. In this case, Eq. (9) becomes

$$(N_i)_j = x_i - \frac{f_j(\mathbf{x}) + \sum_{k=1, k \neq i}^n A_{jk}(X_k - x_k)}{A_{ji}} = x_i - \frac{(Q_i)_j}{A_{ji}}, \quad (10)$$

where $(N_i)_j$ and $(Q_i)_j$ indicate quantities that have been evaluated using element j of \mathbf{y}_i as the pivot in the pivoting preconditioner. The image component N_i , and thus the intersection $N_i \cap X_i$, can be manipulated by choosing different elements j to be the pivot.

For each component, $X_i, i = 1, \dots, n$, of the current interval being tested, $\mathbf{X} = \mathbf{X}^{(k)}$, the procedure implemented in LISS_LP is summarized algorithmically as follows:

1. Initialize: set $j = 1$; $Wx = w[X_i]$; $end_i = 0$. The quantity $w[X_i]$ is the width of X_i .
2. If $0 \notin A_{ji}$ (i.e., it is a C-type pivot), then continue. Otherwise, go to Step 5.
3. Calculate $(N_i)_j$ from Eq. (10), and update $X_i \leftarrow (N_i)_j \cap X_i$.
4. If X_i is empty, return, indicating that the current interval being tested contains no solutions and can be eliminated. Otherwise, work on real-point selection:
 - (a) If $w[X_i] < Wx$ (i.e., X_i has been improved using the pivoting preconditioner), then continue. Otherwise, go to Step 5.
 - (b) Using Eq. (10), compute $X_L = (N_i)_j \cap X_i$ using the trial real point \hat{x} determined by following the left endpoint rule given by Gau and Stadther²² (in step 4.1(b) on p. 834). If X_L is empty, return, indicating that the current interval being tested contains no solutions and can be eliminated. If $\hat{x}_i = \overline{X_i}$, set $end_L = 1$; otherwise, $\hat{x}_i = \underline{X_i}$ and $end_L = 0$.

- (c) Using Eq. (10), compute $X_R = (N_i)_j \cap X_i$ using the trial real point \hat{x} , determined by following the right endpoint rule given by Gau and Stadtherr²² (in step 4.1(c) on p. 834). If X_R is empty, return, indicating that the current interval being tested contains no solutions and can be eliminated. If $\hat{x}_i = \overline{X_i}$, set $\text{end}_R = 1$; otherwise, $\hat{x}_i = \underline{X_i}$ and $\text{end}_R = 0$.
- (d) Perform $X_i \leftarrow X_L \cap X_R$ and if X_i is empty, return, indicating that the current interval being tested contains no solutions and can be eliminated.
- (e) Update $Wx \leftarrow w[X_i]$.
- (f) Choose an endpoint of X_i to use in corner for LP:
- i. If $\text{end}_L = 0$ and $\text{end}_R = 0$ (that is, both left and right endpoint rules pick the lower end of X_i), then $\text{end}_i \leftarrow \text{end}_i - 2$.
 - ii. If $\text{end}_L = 1$ and $\text{end}_R = 1$ (that is, both left and right endpoint rules pick the upper end of X_i), then $\text{end}_i \leftarrow \text{end}_i + 2$.
 - iii. If $\text{end}_L = 1$ and $\text{end}_R = 0$:
 - if $w[X_L] > w[X_R]$ (that is, lower end of X_i is favorable), then $\text{end}_i \leftarrow \text{end}_i - 1$.
 - otherwise, upper end of X_i is favorable, and $\text{end}_i \leftarrow \text{end}_i + 1$.
 - iv. If $\text{end}_L = 0$ and $\text{end}_R = 1$:
 - if $w[X_L] > w[X_R]$ (that is, upper end of X_i is favorable), then $\text{end}_i \leftarrow \text{end}_i + 1$.
 - otherwise, lower end of X_i is favorable, and $\text{end}_i \leftarrow \text{end}_i - 1$.
5. Iterate: if $j < n$, then $j \leftarrow j + 1$ and go to Step 2.
6. If $\text{end}_i > 0$, choose upper endpoint of X_i to use in corner. Otherwise, use lower endpoint in corner.

Note, that the pivoting preconditioning procedure used here (Steps 2 and 3) differs in some aspects from the scheme described by Gau and Stadtherr:²²

1. The pivoting preconditioner is only applied to C-type pivots ($0 \notin A_{ji}$). Gau and Stadtherr²² also apply the pivoting preconditioner in the case of E-type pivots ($0 \in A_{ji}$, $0 \neq A_{ji}$ and $0 \notin (Q_i)_j$), which results in splitting the current box $\mathbf{X}^{(k)}$. Here we have chosen not to use E-type pivots to avoid the additional complexity resulting from the splitting of the current interval.
2. Gau and Stadtherr²² applied Eq. (10) in real arithmetic, using it to determine the pivot j that minimized $w[(N_i)_j \cap X_i]$. Then for this pivot only, the intersection $(N_i)_j \cap X_i$ is computed rigorously using interval arithmetic. Here we compute $(N_i)_j \cap X_i$ with interval arithmetic for each j and in effect determine the intersection of all of these results. This is more expensive to implement than the procedure used by Gau and Stadtherr,²² but may result in some additional reduction in the width of X_i .
3. As seen in Step 4(a), the real-point selection scheme is applied to seek additional shrinkage of X_i , and to help select a corner of \mathbf{X} for use in the LP scheme, only when pivoting preconditioner has resulted in some improvement. Gau and Stadtherr²² apply the real-point selection scheme in some additional circumstances.

These differences can be regarded as an attempt to improve the procedure and tune it for the context of the LP strategy used here, as opposed to the context of an inverse-midpoint preconditioned Gauss-Seidel approach.

Application of the LP strategy requires that a corner of the current box $\mathbf{X}^{(k)}$ be selected and used as the real point $\mathbf{x}^{(k)}$ in Eq. (1). The width of the solution set of Eq. (1) may depend strongly

on the choice of this point. Steps 4(f) and 6 provide a heuristic for trying to determine which end of X_i is most likely, when used as a component of the corner point, to result in a relatively small solution set. For some values of j the upper end of X_i may appear preferable, and for other values of j the lower end may appear preferable. This preference may appear strong, as in the cases treated in Steps 4(f)(i) and 4(f)(ii), or weak, as in the cases treated in Steps 4(f)(iii) and 4(f)(iv). The heuristic deals with potentially contradictory preferences by assigning weights (two for strong preference, one for weak preference) and summing to determine an overall preference.

4.2 Error Bounds for LP Objective Value

The LP strategy outlined above in Section 3 is used to determine exact (within round out) bounds on the solution set of the linear interval equation system, Eq. (1), that must be solved in the context of the interval-Newton method. Since the LP problems are solved using floating point arithmetic on a computer, rounding errors will always occur and could lead to unreliable bounds on the solution set, especially for ill-conditioned problems. To try to ameliorate this potential difficulty, the problem is scaled using the method described by Curtis and Reid.³⁵ However, a rigorous procedure is needed to be sure that the exact bounds on the solution set are enclosed when using the LP strategy. In particular, rigorous upper bounds are needed on the solutions of the LP maximization problems, and rigorous lower bounds are needed on the solutions of the LP minimization problems. A procedure for determining these bounds, based on primal-dual relationships, is given here. Similar procedures have also been recently developed independently by Neumaier and Shcherbina³⁶ and Jansson.³⁷

The linear programming subproblems can be stated in the following standard form:

$$\begin{aligned}
\max \quad & P = \mathbf{c}^T \mathbf{z} \\
\text{s.t.} \quad & \mathbf{A} \mathbf{z} \leq \mathbf{b} \\
& \mathbf{0} \leq \mathbf{z} \leq \mathbf{u},
\end{aligned} \tag{11}$$

where, $\mathbf{A} \in \mathbf{R}^{m \times n}$, $\mathbf{b} \in \mathbf{R}^m$, and $\mathbf{c}, \mathbf{z}, \mathbf{u} \in \mathbf{R}^n$. Associated with the LP problem (11), called the *primal problem*, there is another LP problem, called the *dual problem*:

$$\begin{aligned}
\min \quad & D = \mathbf{b}^T \boldsymbol{\pi} + \mathbf{u}^T \boldsymbol{\omega} \\
\text{s.t.} \quad & \mathbf{A}^T \boldsymbol{\pi} + \boldsymbol{\omega} \geq \mathbf{c} \\
& \boldsymbol{\pi} \geq \mathbf{0}, \quad \boldsymbol{\omega} \geq \mathbf{0},
\end{aligned} \tag{12}$$

where $\boldsymbol{\pi} \in \mathbf{R}^m$ and $\boldsymbol{\omega} \in \mathbf{R}^n$. The duality theorem states that if either problem has a finite optimal solution, then the other also has a finite optimal solution, and that the two optimal objective values are equal in value; that is,

$$\mathbf{c}^T \hat{\mathbf{z}} = \hat{P} = \hat{D} = \mathbf{b}^T \hat{\boldsymbol{\pi}} + \mathbf{u}^T \hat{\boldsymbol{\omega}}, \tag{13}$$

where $\hat{\mathbf{z}}$ and $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\omega}})^T$ denote the optimal solution of primal and dual problem, respectively, and \hat{P} and \hat{D} denote the optimal objective value of primal and dual problems, respectively. Let \mathbf{z}_0 be a feasible solution to the primal and $(\boldsymbol{\pi}_0, \boldsymbol{\omega}_0)^T$ be a feasible solution to the dual; then

$$P_0 = \mathbf{c}^T \mathbf{z}_0 \leq \mathbf{c}^T \hat{\mathbf{z}} = \hat{P} = \hat{D} = \mathbf{b}^T \hat{\boldsymbol{\pi}} + \mathbf{u}^T \hat{\boldsymbol{\omega}} \leq \mathbf{b}^T \boldsymbol{\pi}_0 + \mathbf{u}^T \boldsymbol{\omega}_0 = D_0. \tag{14}$$

This is known as the *weak duality* property. Eq. (14) provides an upper bound on the optimal objective value of the primal problem (maximization problem) in terms of a feasible solution of the dual problem. That is, if a feasible solution of the dual problem, such as $(\boldsymbol{\pi}_0, \boldsymbol{\omega}_0)^T$, can be found, it will provide an upper bound on the solution (optimal objective value, \hat{P}) of the primal

maximization problem. Moreover, in the context of LISS_LP, this allows the solution set of the linear interval equation system to be rigorously enclosed. To provide tight bounds, it is clearly preferable that the feasible solution of the dual problem used to provide the bounds be as close as possible to the optimal solution \hat{D} .

A standard LP routine applied to compute the solution \hat{z} to the primal problem will also compute in the process the optimal values of the dual variables $\hat{\pi}$. Assume that the results actually computed using the LP solver are given by \tilde{x} and $\tilde{\pi}$. Since rounding errors may have occurred in the LP solver, \tilde{x} and $\tilde{\pi}$ should be regarded as approximations of the true solutions \hat{x} and $\hat{\pi}$. Beginning with $\tilde{\pi}$, the following procedure is used to find a feasible solution of the dual problem that is close to the true optimal solution:

1. In principle, all components of π should be nonnegative. However, it is possible that due to numerical errors that this is not true for the computed $\tilde{\pi}$. Thus, define a feasible π_0 that is close to the optimal value, by rounding all negative components of $\tilde{\pi}$ to zero, that is

$$\pi_{0j} = \max\{\tilde{\pi}_j, 0\}, \quad j = 1, \dots, m. \quad (15)$$

2. Compute $\tilde{\omega}$ from the dual constraint,

$$\tilde{\omega} = c - A^T \pi_0. \quad (16)$$

3. Define ω_0 by rounding all negative components of $\tilde{\omega}$ to zero, that is

$$\omega_{0j} = \max\{\tilde{\omega}_j, 0\}, \quad j = 1, \dots, n. \quad (17)$$

Eq. (15) ensures that $\pi_0 \geq \mathbf{0}$; Eq. (17) ensures that $\omega_0 \geq \mathbf{0}$; and from Eq. (16) and (17)

$$\omega_0 \geq \tilde{\omega} = c - A^T \pi_0, \quad (18)$$

that is,

$$\mathbf{A}^T \boldsymbol{\pi}_0 + \boldsymbol{\omega}_0 \geq \mathbf{c}. \quad (19)$$

Thus, the computed $(\boldsymbol{\pi}_0, \boldsymbol{\omega}_0)^T$ satisfies the constraints of the dual problem (12) and indeed is a feasible solution of the dual problem. From Eq. (14), $D_0 = \mathbf{b}^T \boldsymbol{\pi}_0 + \mathbf{u}^T \boldsymbol{\omega}_0$ is an upper bound on \hat{P} , the optimal value of primal maximization problem.

The calculation of an upper bound on \hat{P} in this way requires only few additional matrix-vector operations, which are performed using interval arithmetic to guarantee the enclosure (the lower interval bound does not need to be computed). Note that the above procedure is based on a primal maximization problem. To deal with the minimization problems that must be solved, $\min P$ can be transformed to the maximization problem $\max(-P)$ and the above procedure used to obtain the upper bound D_0 , so that $-\hat{P} \leq D_0$. Thus, $-D_0 \leq \hat{P}$, and a lower bound on the optimal objective value of the minimization problem has been obtained. If the system is ill-conditioned it is possible that the bounds obtained using this procedure will be relatively loose; nevertheless, the bounds will be rigorous.

4.3 LISS_LP Procedure for the Interval-Newton Equation

The overall procedure for using linear programming in solving the interval-Newton equation can be summarized algorithmically as follows:

1. Apply pivoting preconditioner and perform real-point selection as shown in Section 4.1. Determine the corner of the current box to be used in the LP strategy.
2. Scale the problem using the method of Curtis and Reid.³⁵
3. Find a beginning feasible basic solution. If there is no feasible basic solution, return, indicating

that the current interval being tested contains no solutions and can be eliminated. Record all necessary information, which will be used to restore this feasible basic solution as the beginning basic feasible solution for each of subproblems.

4. Using LP, find the optimal solution for each of the $2n$ subproblems. Begin all the subproblems with the same feasible solution, as found in Step 3. The reason for restoring the same beginning feasible basic solution for each subproblem, rather than using the final one from the previous subproblem, is to prevent accumulation of numerical errors and possible numerical instability.
5. Apply the procedure in Section 4.2 to calculate an upper bound on the optimal objective value in the maximization subproblems, and a lower bound on the optimal objective value in the minimization subproblems. This step can be skipped if the optimum occurs at a bound of the current interval. In this case, the original bound value can be used.
6. Scale back to the original scale, and assemble the computed bounds on the solution set of the linear interval system.
7. Return.

The LISS_LP procedure was implemented in C++. Two implementations were developed, one for use on Sun workstations, and another for use on Linux workstations. The Sun version makes use of the Sun Interval Arithmetic Library. The Linux version is portable and uses simulated directed rounding (as in the Fortran 77 INTLIB routines³⁸) in performing interval arithmetic.

5 Results and Discussion

In this section, we present the results of numerical experiments using LISS_LP in implementing the interval-Newton approach for global optimization. To do this we compare LISS_LP to the HP/RP approach of Gau and Stadtherr,²² which itself provides substantial improvements in computational performance over a standard interval-Newton method, and which performs well²² relative to the α -BB approach. Additional examples of the effectiveness of LISS_LP compared to HP/RP approach can also be found in Lin and Stadtherr.¹⁹

The first three test problems are parameter estimation problems, formulated using the error-in-variables (EIV) approach, and demonstrate the case in which the number of variables is relatively large (at least in the context of deterministic global optimization). The fourth test problem demonstrates the case in which there are a very large number of local minima or other stationary points and multiple global minima. The last test problem is a constrained optimization problem arising in an engineering design problem.

Comparisons to the HP/RP approach were performed on a Sun Blade 1000 model 1600 workstation. Other tests were also performed on a Dell Precision workstation 530MT with dual Intel Xeon 1.7GHz processors and 1G RAM under Red Hat Linux 7.3. Only one processor was used during testing.

5.1 Problem 1

This is a parameter estimation problem using the error-in-variables (EIV) approach to estimate kinetic parameters for an irreversible, first order reaction $A \xrightarrow{k_1} B$, using data from an adiabatic

CSTR. The reaction rate constant can be expressed as

$$k_1 = c_1 \exp\left(\frac{-Q_1}{RT}\right), \quad (20)$$

where the Arrhenius constants c_1 and Q_1 are the parameters to be determined. The two parameters are estimated from simulated measurements with 10 data points for 5 state variables, namely the inlet temperature, the outlet temperature, the outlet concentrations of A and B, and the inlet concentrations of A. The problem^{39,40} is formulated as an unconstrained global optimization problem with 22 variables. Complete problem details are given by Gau and Stadtherr.¹⁴

The global optimization problem was solved successfully, with comparative computational performance results shown in Table 1, which shows the number of interval-Newton (I-N) tests performed, and the CPU time in seconds on the Sun Blade workstation. Table 1 shows a comparison of LISS_LP and HP/RP; two versions of LISS_LP are compared. One version, denoted LISS_LP(LM), uses a local minimizer in updating the upper bound on the global minimum, as described in Section 2. The other version does not use a local minimizer in doing this update. In comparison to HP/RP, both versions of LISS_LP provide a large reduction in the number of I-N tests required. This is due to the ability, provided by the LP strategy, to tightly bound the solution set of the interval-Newton equation. However the percent reduction in CPU time is less than the percent reduction in I-N tests, due to the overhead in solving the LP subproblems.

When the local minimizer is used, both the number of I-N tests and CPU time are reduced somewhat since the “better” upper bound results in more intervals being discarded due to the objective range test since they cannot contain the global minimum. This suggests that a more sophisticated local minimizer, as opposed to the the simple direct search method^{28,29} currently implemented, might further improve the computational performance of interval-Newton methods

for global optimization problems.

We have also implemented LISS_LP for a Linux workstation. Table 2 shows the computational performance on the Dell workstation (Intel Xeon processor running Linux). This was done only for the case of LISS_LP with the local minimizer. As should be expected, the only significant difference from the results in Table 1 is that the CPU time is less since this is a faster machine. Continuing improvements in computational speed (increasing at a rate of about two orders of magnitude per decade for a single processor), will make the interval-Newton methodology an increasingly attractive tool for reliable problem solving.

5.2 Problem 2

This is a parameter estimation problem using the EIV approach to estimate the kinetic parameters for chemical conversion of initially pure species A to species B in isothermal batch reactor, $A \xrightarrow{k} B$, as described by Bard.⁴¹ An Arrhenius relationship for the rate constant is again assumed,

$$k = \theta_1 \exp\left(-\frac{\theta_2}{T}\right), \quad (21)$$

and so the parameters θ_1 and θ_2 must be determined. The two parameters are estimated from experimental measurements of the reaction time t , the system temperature, and the fraction of the initial amount of component A that remains at time t . Two sets of data are considered for this system. The first data set (Problem 2-1) consists of 15 data points, and there are 7 data points in the second set (Problem 2-2). The problems are formulated as unconstrained global optimization problems with 32 (Problem 2-1) and 16 (Problem 2-2) independent variables, respectively. Complete problem details are given by Gau and Stadtherr.¹⁴ For the second data set, some of the data points are suspicious, and, to investigate, the objective range test was turned off to obtain all of the stationary points instead of just the global minimum, as Gau and Stadtherr¹⁴ also did.

The two problems were solved successfully, with computational performance results shown in Table 1 and Table 2. In Problem 2-1, although the number of I-N tests has been reduced by a factor of about 4 when using LISS_LP, the CPU time is almost the same for the two methods. Again this is due to the overhead in solving the LP problems. In Problem 2-2, two stationary points are found: one is the reported global minimum with objective value of 16.55240 and the other is a local minimum with objective value of 36.46666. Note that since the objective range test was turned off in Problem 2-2 in order to find all the stationary points, there is no use for the local minimizer on this problem.

5.3 Problem 3

This is a parameter estimation problem using the EIV approach in modeling an isothermal pseudo-differential reactor for the catalytic hydrogenation of phenol on a palladium catalyst, as described by Rod and Hancil.⁴² There are 28 experimental kinetic data points of the partial pressure of phenol (P_1), the partial pressure of hydrogen (P_2), and the initial reaction rate (r). It is desired to fit this kinetic data to a semi-empirical model of the form

$$r = \frac{\theta_1 \theta_2^2 \theta_3 P_1 P_2^2}{(1 + \theta_1 P_1 + \theta_2 P_2)^3}, \quad (22)$$

where θ_1 , θ_2 and θ_3 are the parameters to be estimated. The problem is formulated as an unconstrained global optimization problem with 59 variables, with complete details given by Gau and Stadtherr.¹⁵

The global optimization problem was solved successfully, with computational performance results shown in Table 1 and 2. Due to the relatively large number of variables in this problem, a sparse linear programming routine was implemented in LISS_LP. Both I-N tests and CPU time are reduced compared to HP/RP. This is the largest problem (in terms of number of independent

variables) considered here. However, we have solved some much larger problems, as reported in Lin and Stadtherr,¹⁹ with up to 264 variables. On these larger problems, the impact of the LP overhead appears to be less significant, and the percent reduction in CPU time more closely corresponds to the percent reduction in I-N tests.

5.4 Problem 4

This problem is to find the global minimum of the function

$$f(\mathbf{x}) = 100 \prod_{i=1}^N \sum_{j=1}^5 \left(\frac{j^5}{4425} \cos(j + jx_i) \right) + \frac{1}{N} \sum_{i=1}^N (x_i - x_{0,i})^2, \quad (23)$$

where $x_i \in [x_{0,i} - 20, x_{0,i} + 20]$ and $x_{0,i} = 3$, $i = 1, \dots, N$. This is used as a test problem by Siirola *et al.*⁴³ There are 2048 local minima for the case $N = 2$ and on the order of a hundred million (10^8) local minima for the case $N = 5$. The problem also has multiple (N) global minimizer points. The problems were solved for the cases of $N = 2$ to $N = 6$ on the Dell workstation (Intel Xeon processor running Linux) using LISS_LP with the local minimizer (i.e., using the best of the three methods compared in Problems 1–3).

Results are shown in Table 3. For each value of N , there are N global minimizer points, all of which have been found. The global minimizer points can all be expressed in terms of only two numbers, denoted in Table 3 as x_i^* and $x_{j \neq i}^*$. The i -th global minimizer point will have the value x_i^* for its i -th element, and the value $x_{j \neq i}^*$ for its other $N - 1$ elements.

For the search of the binary tree generated in the bisection process, we generally use an efficient depth-first search procedure. However, for global optimization (branch-and-bound) problems, some type of heuristic “best”-first search might be used to try to reduce the number of boxes that must be tested. In our experience, a difficulty that may arise when this is done is that the number of boxes that remains in the stack to be tested may become very large in comparison to depth-first

search, resulting in much larger memory requirements. One type of best-first search is to choose as the next box to be tested the interval for which the lower bound on the objective function is the smallest.²⁵ When this is done for the case of $N = 5$, the number of I-N tests is reduced from 614,522 to 155,666, the figure reported in Lin and Stadtherr.¹⁹ However, this causes the maximum depth of boxes in the stack to increase to 98,650, as compared to 28 for depth-first search. These trade-offs need to be further investigated. The results in Table 3 are for the depth-first search procedure.

5.5 Problem 5

This problem deals with the maximization of the drying production rate in a through-circulation dryer.^{44,45} The problem is to find the mass flow rate x_1 , and the bed thickness x_2 , such that the drying production rate P is maximized. The problem is formulated as a constrained optimization problem:

$$\begin{aligned} \max_{x_1, x_2} \quad & P = 0.033x_1 (A + 0.095 - B)^{-1} \\ \text{s.t.} \quad & 0.2 - 4.62 \times 10^{-10} x_1^{2.85} x_2 - 1.055 \times 10^{-4} x_1 \geq 0 \\ & 1.75/12 - 8.20 \times 10^{-7} x_1^{1.85} x_2 \geq 0 \\ & 0.64 - 109.6 \frac{x_2}{x_1^{0.41}} (A + 0.095 - B) \geq 0, \end{aligned} \tag{24}$$

where

$$A = \frac{0.036}{1 - e^{-107.9x_2/x_1^{0.41}}}, \tag{25}$$

$$B = \frac{9.27 \times 10^{-4} x_1^{0.41}}{x_2} \ln \left(\frac{1 - e^{-5.39x_2/x_1^{0.41}}}{1 - e^{-107.9x_2/x_1^{0.41}}} \right). \tag{26}$$

The global maximum is sought in the initial interval given by $x_1 \in [10, 10000]$ and $x_2 \in [0.01, 10]$. The problem was solved by applying the interval-Newton approach to solve the Fritz-John condi-

tions. A different local minimizer must be used in this problem to account for the constraints. For this purpose, we used the successive quadratic programming (SQP) method implemented by Chen and Stadtherr.⁴⁶ Using the interval methodology described above (LISS_LP with local minimizer), the global maximum of drying production rate is found to be $P = 172.487$, with $x_1 = 975.831$, and $x_2 = 0.524$. The number of interval-Newton tests was 76 and the total CPU time required was 2.1 seconds on the Dell workstation (Intel Xeon processor running Linux).

6 Concluding Remarks

We have described here the implementation of an LP-based method to solve the linear interval equation system arising in the context of the interval-Newton approach for nonlinear equation solving and global optimization. The method can obtain tighter bounds on the solution set of the linear interval system than the preconditioned interval Gauss-Seidel approach, and thus leads to a large reduction in the number of subintervals that must be tested during the interval-Newton procedure. However, the difference between the overhead required to solve the LP subproblems and that required to perform the preconditioned Gauss-Seidel method may lead to relatively smaller or larger improvements in overall computational time, depending on the size of the problem. With sparse linear algebra in the LP subproblems, the method can be successfully applied to problems involving over two hundred variables.¹⁹

We have demonstrated that the interval-Newton approach is a powerful, deterministic approach for the solution of a number of global optimization problems, such as arise in chemical engineering and other areas of engineering and science. Problems with a very large number of local optima can be effectively solved, as well as problems with a relatively large number of variables. Continuing improvements in methodology, together with advances in software and hardware (e.g., for parallel

computing⁴⁷) will make this an increasingly attractive problem solving tool.

Acknowledgments

This work has been supported in part by the donors of The Petroleum Research Fund, administered by the ACS, under Grant 35979-AC9, and by the Indiana 21st Century Research & Technology Fund.

References

- (1) Stadtherr, M. A.; Schnepper, C. A.; Brennecke, J. F. Robust Phase Stability Analysis Using Interval Methods. *AIChE Symp. Ser.* **1995**, *91*(304), 356.
- (2) McKinnon, K. I. M.; Millar, C. G.; Mongeau M. Global Optimization for the Chemical and Phase Equilibrium Problem Using Interval Analysis. In *State of the Art in Global Optimization: Computational Methods and Applications*; Floudas, C. A.; Pardalos, P.M., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.
- (3) Tessier, S. R.; Brennecke, J. F.; Stadtherr, M. A. Reliable Phase Stability Analysis for Excess Gibbs Energy Models. *Chem. Eng. Sci.* **2000**, *55*, 1785.
- (4) Hua, J. Z.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of Phase Stability Using Interval Analysis: Cubic Equation of State Models. *Comput. Chem. Eng.* **1998**, *22*, 1207.
- (5) Hua, J. Z.; Brennecke, J. F.; Stadtherr, M. A. Enhanced Interval Analysis for Phase Stability: Cubic Equation of State Models. *Ind. Eng. Chem. Res.* **1998**, *37*, 1519.
- (6) Stradi, B. A.; Xu, G.; Brennecke, J. F.; Stadtherr, M. A. Modeling and Design of an Environmentally Benign Reaction Process. *AIChE Symp. Ser.* **2000**, *96*(323), 371–375.
- (7) Xu, G.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of Phase Stability and Equilibrium from the SAFT Equation of State. *Ind. Eng. Chem. Res.* **2002**, *41*, 938–952.
- (8) Stradi, B. A.; Brennecke, J. F.; Kohn, J. P.; Stadtherr, M. A. Reliable Computation of Mixture Critical Points. *AIChE J.* **2001**, *47*, 212–221.

- (9) Maier, R. W.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of Homogeneous Azeotropes. *AIChE J.* **1998**, *44*, 1745.
- (10) Maier, R. W.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of Reactive Azeotropes. *Comput. Chem. Eng.* **2000**, *24*, 1851–1858.
- (11) Xu, G.; Scurto, A. M.; Castier, M.; Brennecke, J. F.; Stadtherr, M. A. Reliable Computation of High Pressure Solid-Fluid Equilibrium. *Ind. Eng. Chem. Res.* **2000**, *39*, 1624–1636.
- (12) Scurto, A. M.; Xu, G.; Brennecke, J. F.; Stadtherr, M. A. Phase Behavior and Reliable Computation of High-Pressure Solid-Fluid Equilibrium with Cosolvents. *Ind. Eng. Chem. Res.* **2003**, in press.
- (13) Gau, C.-Y.; Brennecke, J. F.; Stadtherr, M. A. Reliable Parameter Estimation in VLE Modeling. *Fluid Phase Equilib.* **2000**, *168*, 1–18.
- (14) Gau, C.-Y.; Stadtherr, M. A. Reliable Nonlinear Parameter Estimation Using Interval Analysis: Error-in-Variable Approach. *Comput. Chem. Eng.* **2000**, *24*, 631–638.
- (15) Gau, C.-Y.; Stadtherr, M. A. Deterministic Global Optimization for Error-in-Variables Parameter Estimation. *AIChE J.* **2002**, *48*, 1191–1197.
- (16) Maier, R. W.; Stadtherr, M. A. Reliable Density-Functional-Theory Calculations of Adsorption in Nanoporous Materials. *AIChE J.* **2001**, *47*, 1874–1884.
- (17) Adjiman C. S.; Dallwig, S.; Floudas, C. A.; Neumaier, A. A Global Optimization Method, Alpha-BB, for General Twice-Differentiable Constrained NLPs – I. Theoretical Advances. *Comput. Chem. Eng.* **1998**, *22*, 1137.

- (18) Adjiman C. S.; Androulakis, I. P.; Floudas, C. A. A Global Optimization Method, Alpha-BB, for General Twice-Differentiable Constrained NLPs – II. Implementation and Computational Results. *Comput. Chem. Eng.* **1998**, *22*, 1159.
- (19) Lin, Y.; Stadtherr, M. A. Advances in Interval Methods for Deterministic Global Optimization in Chemical Engineering. *J. Global Optim.* **2003**, in press.
- (20) Kearfott, R. B. Preconditioners for the Interval Gauss-Seidel Method. *SIAM J. Numer. Anal.* **1990**, *27*, 804.
- (21) Kearfott, R. B. Decomposition of Arithmetic Expressions to Improve the Behavior of Interval Iteration for Nonlinear Systems. *Computing* **1991**, *47*, 169.
- (22) Gau, C.-Y.; Stadtherr, M. A. New Interval Methodologies for Reliable Chemical Process Modeling. *Comput. Chem. Eng.* **2002**, *26*, 827–840.
- (23) Jaulin, L.; Kieffer, M.; Didrit, O.; Walter, É. *Applied Interval Analysis*; Springer-Verlag: London, 2001.
- (24) Kearfott, R. B. *Rigorous Global Search: Continuous Problems*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.
- (25) Hansen, E. R. *Global Optimization Using Interval Analysis*; Marcel Dekker: New York, 1992.
- (26) Neumaier, A. *Interval Methods for Systems of Equations*; Cambridge University Press: Cambridge, England, 1990.
- (27) Schnepper, C. A.; Stadtherr, M. A. Robust Process Simulation using Interval Methods. *Comput. Chem. Eng.* **1996**, *20*, 187.

- (28) Hooke, R.; Jeeves, T. A. Direct Search Solution of Numerical and Statistical Problems. *J. Assoc. Comput. Mach.* **1961**, *8*, 212–229.
- (29) Kaupe, A. F. Algorithm 178: Direct Search. *Commun. ACM* **1963**, *6*, 313.
- (30) Rohn, J.; Kreinovich, V. Computing Exact Componentwise Bounds on Solution of Linear Systems with Interval Data Is NP-Hard. *SIAM J. Matrix. Anal.* **1995**, *16*, 415–420.
- (31) Beaumont, O. Solving Interval Linear Systems with Linear Programming Techniques. *Lin. Algebra Appl.* **1998**, *281*, 293–309.
- (32) Jansson, C. Calculation of Exact Bounds for the Solution Set of Linear Interval Systems. *Lin. Algebra Appl.* **1997**, *251*, 321–340.
- (33) Aberth O. The Solution of Linear Interval Equations by a Linear Programming Method. *Lin. Algebra Appl.* **1997**, *259*, 271–279.
- (34) Oettli, W.; Prager, W. Compatibility of Approximate Solution of Linear Equation with Given Error Bounds for Coefficients and Right-Hand Sides. *Numerische Mathematik* **1964**, *6*, 405–408.
- (35) Curtis, A. R.; Reid, J. K. On the Automatic Scaling of Matrices for Gaussian Elimination. *J. Inst. Maths. Applics.* **1972**, *10*, 118–124.
- (36) Neumaier, A.; Shcherbina, O. Safe Bounds in Linear and Mixed-Integer Programming. *Math. Prog.* **2003**, in press.
- (37) Jansson, C. A Rigorous Lower Bound for the Optimal Value of Convex Optimization Problems. *J. Global Optim.* **2003**, in press.

- (38) Kearfott, R. B.; Dawande, M.; Du, K.-S.; Hu, C.-Y. Algorithm 737: INTLIB, A Portable FORTRAN 77 Interval Standard Function Library. *ACM Trans. Math. Software* **1994**, *20*, 447–459.
- (39) Esposito, W. R.; Floudas, C. A. Global Optimization in Parameter Estimation of Nonlinear Algebraic Models Via the Error-in-Variables Approach. *Ind. Eng. Chem. Res.* **1998**, *37*, 1841–1858.
- (40) Kim, I.; Leibman, M.; Edgar, T. Robust Error-in-Variables Estimation Using Nonlinear Programming Techniques. *AIChE J.* **1990**, *36*, 985–993.
- (41) Bard, Y. *Nonlinear Parameter Estimation*; Academic Press: New York, 1974.
- (42) Rod, V.; Hancil, V. Numerical Methods for Estimating Parameters in Nonlinear Models with Errors in the Variables. *Technometrics* **1980**, *27*, 33.
- (43) Siirola, J. D.; Hauen, S.; Westerberg, A. W. Agent-based Strategies for Multiobjective Optimization. Paper 265g, Presented at AIChE Annual Meeting, Indianapolis, IN, November 3–8, 2002.
- (44) Chung, F. S. Mathematical Model and Optimization of Drying Process for a Through-Circulation Dryer. *Can. J. Chem. Eng.* **1972**, *50*, 657–662.
- (45) Chung, F. S. Editorial Letter. *Can. J. Chem. Eng.* **1973**, *51*, 262.
- (46) Chen, H. S.; Stadtherr, M. A. A Modification of Powell’s Dogleg Method for Solving Systems of Nonlinear Equations. *Comput. Chem. Eng.* **1981**, *5*, 143.

- (47) Gau, C.-Y.; Stadtherr, M. A. Dynamic Load Balancing for Parallel Interval-Newton Using Message Passing. *Comput. Chem. Eng.* **2002**, *26*, 811–825.

Table 1: Computational performance of LISS_LP and HP/RP on a Sun Blade 1000 Model 1600. In Problem 2-2, all local minima are sought (not just the global minimum). Thus, the objective range test is turned off, and the local minimizer (LM) strategy is not applicable (NA).

Problem	Variables(n)	HP/RP		LISS_LP		LISS_LP(LM)	
		I-N Tests	CPU time	I-N Tests	CPU time	I-N Tests	CPU time
1	22	9,505	24.0	2,004	19.2	1,258	12.7
2-1	32	144,833	976.2	30,560	992.6	24,817	837.2
2-2	16	69,421	91.8	18,715	68.0	NA	NA
3	59	55,255	2315.9	10,689	1863.3	9,757	1692.4

Table 2: Computational performance of LISS_LP on Dell Precision workstation (Intel Xeon 1.7 GHz running Linux). The local minimizer is used except on Problem 2-2.

Prob.	Variables	I-N Tests	CPU time (s)
1	22	1,237	5.99
2-1	32	24,851	238.48
2-2	16	17,027	26.15
3	59	9,647	726.2

Table 3: Global solution of Sirola’s problem,⁴³ with computational performance on a Dell Precision workstation (Intel Xeon 1.7 GHz running Linux).

Global Minimizer Points					
N	x_i^*	$x_{j \neq i}^*$	Global Minimum	I-N tests	CPU time (s)
2	4.6198510288	5.2820519601	-88.1046253312	807	0.07
3	4.6201099154	5.2824296177	-87.6730486951	10,591	2.12
4	4.6202393815	5.2826184940	-87.4572049443	86,796	33.95
5	4.6203170683	5.2827318347	-87.3276809494	614,522	413.61
6	4.6203688625	5.2828074014	-87.2413242244	4,296,083	4566.42