*Journal of Global Optimization* **29:** 281–296, 2004.

# Advances in Interval Methods for Deterministic Global Optimization in Chemical Engineering

Y. LIN and M.A. STADTHERR
*Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556,USA (E-mail: markst@nd.edu)*

(Received 31 July 2003; accepted in revised form 7 August 2003)

**Abstract.** In recent years, it has been shown that strategies based on an interval-Newton approach can be used to reliably solve a variety of nonlinear equation solving and optimization problems in chemical process engineering, including problems in parameter estimation and in the computation of phase behavior. These strategies provide a mathematical and computational guarantee either that all solutions have been located in an equation solving problem or that the global optimum has been found in an optimization problem. The primary drawback to this approach is the potentially high computational cost. In this paper, we consider strategies for bounding the solution set of the linear interval equation system that must be solved in the context of the interval-Newton method. Recent preconditioning techniques for this purpose are reviewed, and a new bounding approach based on the use of linear programming (LP) techniques is presented. Using this approach it is possible to determine the desired bounds exactly (within round out), leading to significant overall improvements in computational efficiency. These techniques will be demonstrated using several global optimization problems, with focus on problems arising in chemical engineering, including parameter estimation and molecular modeling. These problems range in size from under ten variables to over two hundred, and are solved deterministically using the interval methodology.

**Key words:** Deterministic global optimization, Interval analysis, Linear programming, Molecular modeling, Parameter estimation, Preconditioning

## 1. Introduction

At the core of many mathematical modeling problems is the need to solve a nonlinear model, or to optimize a nonlinear function subject to constraints. In either case, there may arise issues with the reliability of the problem-solving method used. For example, if there are multiple solutions to the model, have all been located? If there are multiple local optima, has the global solution been found? Interval mathematics can provide the modeler with the tools needed to resolve these issues with mathematical and computational certainty, thus providing a degree of problem-solving reliability not available when using standard methods. 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 nonlinear equation solving and optimization problems in chemical engineering, including computation of fluid phase equilibrium from activity coefficient models [21,25], cubic equation-of-state (EOS) models [8,9,23], and statistical associating fluid theory [29], calculation of critical points from cubic EOS models [24], location of azeotropes [15] and reactive azeotropes [16], computation of solid-fluid equilibrium [28], parameter estimation using standard least squares [4] and error-in-variables [3,6], and calculation of adsorption in nanoscale pores from a density functional theory model [17]. 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.

The primary drawback to this approach is the potentially high computational cost. One way to improve the efficiency of the interval-Newton approach for solving a nonlinear equation system is to more tightly bound the solution set of the linear interval equation system that is at the core of this approach. In this paper, we review recent preconditioning techniques [5,11,12], for this purpose, and a new bounding strategy based on the use of linear programming (LP) techniques is presented. 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 this 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. These techniques will be demonstrated using several global optimization problems, with focus on problems arising in chemical engineering, including parameter estimation and molecular modelling.

## 2. Interval analysis

Several good introductions to interval computations are available [7,10,13,18]. Of particular interest here is the interval-Newton method. Given an $n \times n$ nonlinear equation system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ 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 $\phi(\boldsymbol{x})$, a common approach is to seek stationary points, that is, to solve the nonlinear equation system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\nabla}\phi(\boldsymbol{x}) = \boldsymbol{0}$. 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.

Given some initial interval $\boldsymbol{X}^{(0)}$, the interval-Newton solution algorithm is applied to a sequence of subintervals. For a subinterval $\boldsymbol{X}^{(k)}$ in the sequence, the first step is the *function range test*. An *interval extension* $\boldsymbol{F}(\boldsymbol{X}^{(k)})$ of the function

$\boldsymbol{f}(\boldsymbol{x})$ 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 $\boldsymbol{F}(\boldsymbol{X}^{(k)})$ that does not include zero, then the interval can be discarded, since no solution of $\boldsymbol{f}(\boldsymbol{x})=\mathbf{0}$ can exist in this interval. The next subinterval in the sequence may then be considered. Otherwise, testing of $\boldsymbol{X}^{(k)}$ continues.

For a global minimization problem, the next step is the *objective range test*. The interval extension $\Phi(\boldsymbol{X}^{(k)})$, containing the range of $\phi(\boldsymbol{x})$ over $\boldsymbol{X}^{(k)}$ is computed. If the lower bound of $\Phi(\boldsymbol{X}^{(k)})$ is greater than a known upper bound on the global minimum, then $\boldsymbol{X}^{(k)}$ can be discarded since it cannot contain the global minimum and need not be further tested. In cases that 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:

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

is solved for a new interval $\boldsymbol{N}^{(k)}$, where $F'(\boldsymbol{X}^{(k)})$ is an interval extension of the Jacobian of $\boldsymbol{f}(\boldsymbol{x})$, and $\boldsymbol{x}^{(k)}$ is an arbitrary real point in $\boldsymbol{X}^{(k)}$. It has been shown (e.g. [7,13,18]) that any root contained in $\boldsymbol{X}^{(k)}$ is also contained in the *image* $\boldsymbol{N}^{(k)}$. This implies that if the intersection between $\boldsymbol{X}^{(k)}$ and $\boldsymbol{N}^{(k)}$ is empty, then no root exists in $\boldsymbol{X}^{(k)}$, and also suggests the iteration scheme $\boldsymbol{X}^{(k+1)}=\boldsymbol{X}^{(k)}\cap\boldsymbol{N}^{(k)}$. In addition, it has also been shown (e.g. [7,13,18]) that, if $\boldsymbol{N}^{(k)}\subset\boldsymbol{X}^{(k)}$, then there is a *unique* root contained in $\boldsymbol{X}^{(k)}$ and thus in $\boldsymbol{N}^{(k)}$. Thus, after computation of $\boldsymbol{N}^{(k)}$ from Equation (1), there are three possibilities: (1) $\boldsymbol{X}^{(k)}\cap\boldsymbol{N}^{(k)}=\emptyset$, meaning there is no root in the current interval $\boldsymbol{X}^{(k)}$ and it can be discarded; (2) $\boldsymbol{N}^{(k)}\subset\boldsymbol{X}^{(k)}$, meaning that there is *exactly* one root in the current interval $\boldsymbol{X}^{(k)}$; (3) neither of the above, meaning that no conclusion can be drawn. In the last case, if $\boldsymbol{X}^{(k)}\cap\boldsymbol{N}^{(k)}$ is sufficiently smaller that $\boldsymbol{X}^{(k)}$, 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 of subintervals to be tested. 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 $\boldsymbol{f}(\boldsymbol{x})=\mathbf{0}$ have been tightly enclosed or it has been determined that no root exists.

Clearly, the solution of the linear interval system given in Equation (1) is essential to this approach. To see the issues involved in solving such a system, consider the general linear interval system $Az=\boldsymbol{B}$, where the matrix $A$ and the right-hand side vector $\boldsymbol{B}$ are interval-valued. The solution set $S$ of this system is defined by $S=\{z|\tilde{A}z=\boldsymbol{b},\tilde{A}\in A,\boldsymbol{b}\in\boldsymbol{B}\}$. 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 $\boldsymbol{Z}$ containing $S$. Computing the interval hull (the tightest interval containing $S$) is NP-hard [20], but there are several methods for determining an interval $\boldsymbol{Z}$ that contains but overestimates $S$. Various interval-Newton methods differ in how they solve Equation (1) for $\boldsymbol{N}^{(k)}$ 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 $\boldsymbol{N}^{(k)}$ the volume of $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$ is reduced, and it is also more likely that either $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)} = \emptyset$ or $\boldsymbol{N}^{(k)} \subset \boldsymbol{X}^{(k)}$ 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, $\boldsymbol{N}^{(k)}$ is computed component-wise using an interval Gauss–Seidel approach, preconditioned with an inverse-midpoint matrix, as described briefly below. Though the inverse-midpoint preconditioner is a good general-purpose preconditioner, it is not always the most effective approach [11,13]. Recently, a hybrid preconditioning approach, which combines a simple pivoting preconditioner with the standard inverse-midpoint scheme, has been proposed by Gau and Stadtherr [5] and shown to achieve substantially more efficient computational performance than the inverse-midpoint preconditioner alone. This approach is reviewed in the next section, since it will be used as the basis for performance comparisons in later examples (Section 5). However, it still may not yield the tightest enclosure of the solution set, which, as noted above, is in general an NP-hard problem. In Section 4, a linear programming strategy will be applied to solve the linear interval system, Equation (1). Using this approach, exact component-wise bounds on the solution set required in the context of the interval-Newton method can be calculated, while avoiding exponential time complexity. A similar LP strategy has also been proposed for use in the context of certain types of constraint satisfaction problems [10].

## 3. Preconditioning approaches for interval-Newton

In this section, we review preconditioning approaches for solving the interval-Newton equation, Equation (1). First the standard interval Gauss–Seidel, with inverse midpoint preconditioner, is reviewed, and then the hybrid preconditioning approach of Gau and Stadtherr [5] is discussed.

### 3.1. INTERVAL GAUSS–SEIDEL

The interval Gauss–Seidel procedure is frequently used to solve Equation (1) for the image $\boldsymbol{N}^{(k)}$. The interval-Newton equation is first preconditioned using a real matrix $Y^{(k)}$. The preconditioned linear interval equation system can then be expressed as

$$Y^{(k)}F'(\boldsymbol{X}^{(k)})(\boldsymbol{N}^{(k)}-x^{(k)})=-Y^{(k)}\boldsymbol{f}(\boldsymbol{x}^{(k)}). \tag{2}$$

The preconditioner $Y^{(k)}$ used here is commonly taken to be an *inverse-midpoint preconditioner*, which may be either the inverse of the real matrix formed from the midpoints of the elements of the interval Jacobian $F'(\boldsymbol{X}^{(k)})$, or the inverse of the real matrix determined by evaluating the point Jacobian $f'(\boldsymbol{x})$ at the midpoint of $\boldsymbol{X}^{(k)}$.

Defining $\boldsymbol{y}_i$ as the $i$th row of the preconditioning matrix and $\boldsymbol{A}_i$ as the $i$th column of the interval Jacobian $F'(\boldsymbol{X}^{(k)})$, then beginning with $\boldsymbol{X}=\boldsymbol{X}^{(k)}$, the interval Gauss–Seidel scheme used in connection with interval-Newton methods proceeds component by component according to

$$N_i = x_i - \frac{\boldsymbol{y}_i\boldsymbol{f}(\boldsymbol{x}) + \sum_{j=1,j\neq i}^{n} \boldsymbol{y}_i\boldsymbol{A}_j(X_j - x_j)}{\boldsymbol{y}_i\boldsymbol{A}_i} \tag{3}$$

where $\boldsymbol{y}_i\boldsymbol{A}_j$ indicates the inner product of the real row vector $\boldsymbol{y}_i$ and the interval column vector $\boldsymbol{A}_j$, and then,

$$X_i \leftarrow X_i \cap N_i \tag{4}$$

for $i=1,\ldots,n$. Note that after component $N_i$ of the image is calculated from Equation (3) that the intersection in Equation (4) is immediately performed, and the updated $X_i$ then used in the computation of subsequent components of the image. This means that this procedure actually does not enclose the full solution set of Equation (1). but only the part of the solution set necessary for the interval-Newton iteration. Generally, only one pass is made through Equations (3–4) and so after all the $X_i$ have been updated the result is $\boldsymbol{X}=\boldsymbol{X}^{(k+1)}$, the next interval-Newton iterate.

### 3.2. HYBRID PRECONDITIONING APPROACH

The inverse-midpoint preconditioner is a good general-purpose preconditioner. However, as demonstrated by Kearfott [11], it is not always the most effective approach. A hybrid approach in which a simple pivoting preconditoner is used in combination with the standard inverse-midpoint scheme was thus considered by Gau and Stadtherr [5]. Their goal was to develop an approach that would significantly reduce the number of subintervals that must be tested in the interval-Newton algorithm, but that could be implemented with little computational overhead, so that large savings in computation time could be realized.

In a pivoting preconditioner [12], only one element of the preconditioning row $\boldsymbol{y}_i$, called the *pivot* element, is nonzero, and it is assigned a value of one. In this case, Equation (3) becomes

$$(N_i)_j = x_i - \frac{f_j(\boldsymbol{x}) + \sum_{k=1,k\neq i}^{n} A_{jk}(X_k - x_k)}{A_{ji}} \tag{5}$$

where the notation $(\cdot)_j$ is used to indicate a quantity that has been evaluated using element $j$ of $\boldsymbol{y}_i$ 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. An appropriate criterion for choosing $j$ is to seek a preconditioning row $\boldsymbol{y}_i$ that minimizes the width of $N_i \cap X_i$, that is, to solve

$$\max_j \mathrm{w}[(N_i)_j \cap X_i] \tag{6}$$

Since the endpoints of $(N_i)_j$ are easily computed [5] from Equation (5), this objective function is also easy to compute. In Gau and Stadtherr's procedure [5], a pivot $j$ that results in $(N_i)_j \cap X_i = \emptyset$ is first sought. This is called a *discarding-optimal* preconditioner, since $(N_i)_j \cap X_i = \emptyset$ means that the current interval $\boldsymbol{X}$ being tested can now be discarded as containing no root. If there is no such pivot, then the pivot $j$ solving the minimum-width optimization problem given in Equation (6) is chosen. This is called a *contraction-optimal* preconditioner.

As already noted, use of the inverse-midpoint preconditioner does not always result in good performance of the interval-Newton algorithm. The use of a pivoting preconditioner alone does not always lead to good performance either. Thus, Gau and Stadtherr [5] adopted a hybrid strategy. The basic idea is that, in determining a preconditioning row $\boldsymbol{y}_i$, a discarding-optimal pivoting preconditioner is first searched for, and while doing so, the information needed to pick a contraction-optimal pivoting preconditioner is also determined. If in fact there is no discarding-optimal pivoting preconditioner, then either the contraction-optimal pivoting preconditioner or the inverse-midpoint preconditioner is used, depended on which yields the smaller width of $N_i \cap X_i$. Note that, in determining the endpoints of $(N_i)_j$ for purposes of choosing a pivoting preconditioner according to Equation (6), the computations can be done cheaply by using real (not interval) arithmetic. Only when the chosen preconditioning row is actually implemented, in Equation (3), does interval arithmetic actually need to be used.

The complete algorithm for implementing the hybrid preconditioning scheme is given in detail by Gau and Stadtherr [5]. Their numerical experiments show that this approach leads to substantial reductions in computing time requirements, in many cases by multiple orders of magnitude. They also considered another approach for improving the computational efficiency of the interval-Newton method. This involves adjusting the value of the real point $\boldsymbol{x}^{(k)} \in \boldsymbol{X}^{(k)}$ in Equation (1) to further reduce the width of $(N_i)_j \cap X_i$ during the selection of the pivoting preconditioner ($x^{(k)}$, though typically chosen to be the midpoint of $\boldsymbol{X}^{(k)}$, can be any point in $\boldsymbol{X}^{(k)}$). In their numerical experiments, Gau and Stadtherr [5] showed that this combined hybrid preconditioning/real-point adjustment (HP/RP) scheme yielded further reductions in computation time compared to the hybrid preconditioning scheme alone. This HP/RP scheme will be used as the basis for performance comparisons in later examples (Section 5).

## 4. LP strategy for interval-Newton

Consider again the linear interval system $Az=\boldsymbol{B}$. Oettli and Prager [19] show that the solution set $S$ is determined by the constraints

$$|\hat{A}z-\hat{\boldsymbol{B}}|\leqslant \Delta A|z|+\Delta\boldsymbol{B}, \tag{7}$$

where $\hat{A}$ is the component-wise midpoint matrix of the interval matrix $A$, $\Delta A$ is the component-wise half-width (radius) matrix of $A$, and similarly $\hat{\boldsymbol{B}}$ and $\Delta\boldsymbol{B}$ are the midpoint and radius $\boldsymbol{B}$. Equation (7) 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 operation can be dropped, For a given orthant, define the diagonal matrix $D_\alpha$ by

$$(D_\alpha)_{jj}=\begin{cases}1 & z_j\geqslant 0\\ -1 & z_j\leqslant 0\end{cases}\quad j=1,2,\ldots,n. \tag{8}$$

Thus, in each orthant $|z|=D_\alpha z$ and $z=D_\alpha|z|$, and Equation (7) becomes

$$|\hat{A}z-\hat{\boldsymbol{B}}|\leqslant \Delta AD_\alpha z+\Delta\boldsymbol{B} \tag{9}$$

This can be rearranged to the set of linear inequalities

$$\begin{pmatrix}\hat{A}-\Delta AD_\alpha\\ -\hat{A}-\Delta AD_\alpha\end{pmatrix}z\leqslant\begin{pmatrix}\overline{\boldsymbol{B}}\\ -\underline{\boldsymbol{B}}\end{pmatrix}, \tag{10}$$

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

$$\max_z z_j,\quad j=1,2,\ldots,n \tag{11}$$

$$\min_z z_j,\quad j=1,2,\ldots,n \tag{12}$$

each with the $2n$ linear inequality constraints given by Equation (10). 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 Equation (1) that intersects $\boldsymbol{X}^{(k)}$ needs to be found. Consider the choice of the real point $\boldsymbol{x}^{(k)}$ in Equation (1). Here $\boldsymbol{x}^{(x)}$ is an arbitrary point in $\boldsymbol{X}^{(k)}$ typically taken to be the midpoint. However, if $\boldsymbol{x}^{(k)}$ is chosen to be a corner of $\boldsymbol{X}^{(k)}$ instead, the part of the solution set for $\boldsymbol{N}^{(k)} - \boldsymbol{x}^{(k)}$ of Equation (1) that intersects $\boldsymbol{X}^{(k)}$ lies in just one orthant. Thus, in the context of interval-Newton, only $2n$ LP subproblems, each with $2n$ constraints, need to be solved. Furthermore, the LP subproblems have properties that can be exploited. First, all the $2n$ subproblems share the same constraints; that is, 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 can be used to replace the inverse-midpoint preconditioned Gauss–Seidel method for solving Equation (1). It can be combined with the low basis pivoting preconditioner and optimal real-point scheme [5] discussed above to achieve best performance. The pivoting preconditioner can also help LISS_LP to select one of the corners of $\boldsymbol{X}^{(k)}$ to use as the real point, the choice of which may have significant impact on the overall performance of LISS_LP. Since the LP subproblems are solved using floating point arithmetic in the current implementation of LISS_LP, which may cause rounding error concerns, a practical error bound estimator on the solution of the linear system was adopted to ensure the reliability of the solution.

## 5. Results and discussion

In this section, we present the results of numerical experiments to test the effectiveness of 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 [5], which, as noted above, provides substantial improvements in computational performance over a standard interval-Newton method. The first two 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 remaining three problems demonstrate the case in which there are a large number of local minima or other stationary points, including a problem with multiple global minima, and a molecular modelling problem requiring the location of all stationary points on a potential energy surface (PES). All tests were performed on a Sun Blade 1000 model 1600 workstation.

*Table 1.* Computational performance on parameter estimation in VLE modelling (Problem 1)

| | HP/RP | LISS_LP |
|---|---|---|
| I-N tests | 303,589 | 156,182 |
| CPU time | 664.4 | 496.7 |

### 5.1. PROBLEM 1

This is a parameter estimation problem using the EIV approach to estimate parameters in the van Laar equation for activity coefficients. These two parameters are estimated from vapour-liquid equilibrium (VLE) data for the binary system of methanol and 1,2-dichloroethane. The experimental data consist of five experimental data points for four measured state variables, namely pressure, temperature, and liquid- and vapor-phase mole fraction of methanol. The problem [2,14] is formulated as an unconstrained global optimization problem with 12 variables, as explained in [3].

The global optimization problem was solved successfully, with computational performance results shown in Table 1, where the number of interval-Newton (I-N) tests performed, and the CPU time in seconds are given. When LISS_LP is applied, the number of I-N tests is substantially reduced, indicating the effectiveness of LISS_LP in reducing the number of intervals that must be tested. Essentially, by reducing the size of $N^{(k)}$, LISS_LP is able to more quickly identify and discard intervals that contain no solution. However, the percent reduction in overall CPU time is less than the percent reduction in I-N tests. This occurs due to the overhead in solving the LP subproblems.

### 5.2. PROBLEM 2

This is a parameter estimation problem using the EIV approach to estimate the rating parameters for a steady-state heat exchanger network, which consists of four heat exchangers. The four parameters can be estimated from experimental measurements, including six flow measurements and thirteen temperature measurements. Five versions of the parameter estimation problems were solved, differing in the number of data points, which ranges from $m=4$ to $m=20$. In the optimization problem, the number of independent variables is $13m+4$, and thus ranges from 56 to 264. Detail of the problem [1] can be found in [6].

The global optimization problems were solved successfully, with computational performance results shown in Table 2. Due to the large number of variables, sparse linear programming routines were implemented in LISS_LP for these problems.

*Table 2.* Computational performance on parameter estimation in heat exchanger network modelling (Problem 2)

| Data points | Variables | HP/RP | | LISS_LP | |
|---|---|---|---|---|---|
| m | n | I-N tests | CPU time ($s$) | I-N tests | CPU time ($s$) |
| 4 | 56 | 1 | 0.12 | 2 | 0.27 |
| 8 | 108 | 375 | 211.8 | 44 | 38.1 |
| 12 | 160 | 363 | 498.6 | 299 | 346.0 |
| 16 | 212 | 188 | 645.8 | 83 | 316.8 |
| 20 | 264 | 220 | 1357.3 | 81 | 504.9 |

Both I-N tests and CPU time are substantially reduced when $m$ is large, indicating the effectiveness of LISS_LP. A 264-variable deterministic global optimization problem is solved in only about 500 s of CPU time.

### 5.3. PROBLEM 3

This is a global optimization problem given by Trefethen [26] as part of a set of challenge problems in which at least 10 digits of precision were required in the final results. The global minimum of the function

$$\begin{aligned} f(x,y) = {} & \exp(\sin(50x))+\sin(60\exp(y))+\sin(70\sin(x))+\sin(\sin(80y)) \\ & -\sin(10(x+y))+(x^2+y^2)/4 \end{aligned} \tag{13}$$

is sought, where $x\in[-1,1]$ and $y\in[-1,1]$. On the unit square ($[0,1]\times[0,1]$) alone, the function has 667 local minima, as well as many other stationary points.

This global optimization problem was solved successfully, with more than 10 digits of precision, with computational performance results as shown in Table 3. This proves to be a very easy problem to solve using the interval approach. Though the number of I-N tests is reduced when LISS_LP is used, this is balanced on this easy problem by the overhead required to solve the LP problems, and thus there is no overall reduction in CPU time. It appears that, on relatively easy problems, the LP-based strategy is not needed, but still can be used without significant loss of efficiency due to LP overhead.

*Table 3.* Computational performance on Trefethen challenge (Problem 3)

| | HP | LISS_LP |
|---|---|---|
| I-N tests | 1814 | 1179 |
| CPU time | 0.15 | 0.16 |

*Table 4.* Computational performance on Siirola et al. function (Problem 4)

| | Global Minima | I-N tests | CPU time ($s$) |
|---|---|---|---|
| LISS_LP | 5 | 155,666 | 389.17 |
| HP | 5 | 171,918 | 636.76 |

## 5.4. PROBLEM 4

This is to find the global minimum of the function

$$f(x) = 100 \prod_{i=1}^{N} \sum_{j=1}^{5} \left( \frac{j^5}{4425} \cos(j + jx_i) \right) + \frac{1}{N} \sum_{i=1}^{N} (x_i - x_{0,i})^2 \tag{14}$$

where $x_i \in [x_{0,i} - 20, x_{0,i} + 20]$ and $x_{0,i} = 3, i = 1, \ldots, N$. This is used as a test problem by Siirola et al. [22]. The problem is solved here for the case of $N = 5$, for which there are on the order of a hundred million ($10^8$) local minima. Five global minima exist and are successfully located, with the computational performance results shown in Table 4. For the number of variables in this problem, the overhead in solving LP subproblems is actually less than the overhead in doing preconditioned Gauss–Seidel. Therefore, even though the number of I-N tests is reduced by only about 10% when LISS_LP is used, the overall computational time is reduced by almost 40%.

## 5.5. PROBLEM 5

In this final set of examples, we applied the LISS_LP approach to the problem of finding all stationary points of potential energy surfaces for several triatomic molecules [27]. This is useful in the study of reaction pathways and transition states. The Murrel–Sorbie analytic potential energy surface was used. For a triatomic molecule ABC, the geometry of the molecular structure can be described by the three interatomic distances $R_{\mathrm{AB}}$, $R_{\mathrm{AC}}$, and $R_{\mathrm{BC}}$. The potential energy surface

(PES) is then represented by a function of the form $V(R_{\mathrm{AB}}, R_{\mathrm{AC}}, R_{\mathrm{BC}})$. To find all the stationary points of this surface then requires finding all solutions of the stationary conditions

$$\partial V/\partial R_{\mathrm{AB}} = 0 \tag{15}$$

$$\partial V/\partial R_{\mathrm{AC}} = 0 \tag{16}$$

$$\partial V/\partial R_{\mathrm{BC}} = 0 \tag{17}$$

subject to the triangle inequality constraints

$$R_{\mathrm{AB}} \leqslant R_{\mathrm{AC}} + R_{\mathrm{BC}} \tag{18}$$

$$R_{\mathrm{AC}} \leqslant R_{\mathrm{AB}} + R_{\mathrm{BC}} \tag{19}$$

$$R_{\mathrm{BC}} \leqslant R_{\mathrm{AB}} + R_{\mathrm{AC}}, \tag{20}$$

only one of which can be simultaneously active.

When one of the constraints, Equations (18–20), is active then the molecule is collinear. For example, if the molecule is collinear with atom B is in the middle, then $R_{\mathrm{AC}} = R_{\mathrm{AB}} + R_{\mathrm{BC}}$. In this case, the collinearity condition (active constraint) can be used to eliminate one variable in terms of the other two. Now, for the case of B in the middle, the stationarity conditions that must be solved are

$$\frac{\partial}{\partial R_{\mathrm{AB}}} V(R_{\mathrm{AB}}, R_{\mathrm{AB}} + R_{\mathrm{BC}}, R_{\mathrm{BC}}) = 0 \tag{21}$$

$$\frac{\partial}{\partial R_{\mathrm{BC}}} V(R_{\mathrm{AB}}, R_{\mathrm{AB}} + R_{\mathrm{BC}}, R_{\mathrm{BC}}) = 0 \tag{22}$$

with the triangle inequality constraints no longer applicable. There are two other similar equation systems for the cases of atom A in the middle and C in the middle.

In the nonlinear case in which none of the triangle inequality constraints is active, then the three stationarity conditions, Equations (15–17), can be solved and any roots that do not satisfy the inequality constraints discarded. However, we instead put the constraints to use in a constraint propagation (bound contraction) scheme, which is done before the function range test. For example, considering the constraint on $R_{\mathrm{AB}}$ if $\overline{R_{\mathrm{AB}}} > \underline{R_{\mathrm{AC}}} + \underline{R_{\mathrm{BC}}}$, then the upper bound $\overline{R_{\mathrm{AB}}}$ can be contracted to $\overline{R_{\mathrm{AB}}} = \underline{R_{\mathrm{AC}}} + \underline{R_{\mathrm{BC}}}$. Furthermore, if $\underline{R_{\mathrm{AB}}} > \overline{R_{\mathrm{AC}}} + \overline{R_{\mathrm{BC}}}$, then the current subinterval can be discarded since it can not satisfy the inequality constraints. Similar procedures are also applied based on the constraints on $R_{\mathrm{AC}}$ and $R_{\mathrm{BC}}$. Using this

*Table 5.* Summary of performance of LISS_LP on triatomic problems (Problem 5)

| Problem | Stationary points found | CPU time ($s$) |
|---|---|---|
| HCN | 10 | 6.66 |
| HSiN | 11 | 1.07 |
| $CS_2$ | 5 | 2.18 |
| HBO(PES1) | 4 | 0.88 |
| HBO(PES2) | 4 | 0.56 |

*Table 6.* Stationary states for HCN (Problem 5)

| Type | Energy ($eV$) | $R_{CN}$ (Å) | $R_{CH}$(Å) | $R_{NH}$(Å) |
|---|---|---|---|---|
| Minimum | −5.548223 | – | 2.332871 | 1.038900 |
| Saddle | 8.094668 | – | 0.857572 | 0.806900 |
| Minimum | −12.972507 | 1.159150 | – | 0.993336 |
| Minimum | −13.592215 | 1.153198 | 1.065498 | – |
| Saddle | −5.249952 | 2.344235 | 2.980408 | 1.044278 |
| Saddle | −1.937592 | 2.311895 | 1.792854 | 2.327696 |
| Saddle | −3.102483 | 2.582864 | 1.081559 | 2.737335 |
| Saddle | −11.444169 | 1.117973 | 1.053919 | 1.387750 |
| Saddle | −11.345398 | 0.929065 | 1.039138 | 1.041348 |
| Minimum | −11.379410 | 0.857321 | 0.980845 | 0.989052 |

scheme, stationary points that do not satisfy the inequality constraints are never found.

To solve the overall problem of finding all the stationary points of the PES, one must actually solve four subproblems, corresponding to the three collinear cases and the one noncollinear case. The LISS_LP approach was used successfully to find all stationary points for the molecules HCN, HSiN, $CS_2$, and HBO. Details of the analytic potential energy surface for each molecule can be found in [27]. Note that two different potential energy surfaces for HBO were used. Table 5 shows the performance of LISS_LP for each of the problems, and the results for the stationary points are summarized in Tables 6–10, where "–" indicates the long side of a collinear molecule; these results are consistent with those found in [27] using a different approach. The initial search region for each distance was [0.7,5.0]Å, as in [27]. The interval approach using LISS_LP is clearly very effective in solving these problems.

*Table 7.* Stationary states for HSiN (Problem 5)

| Type | Energy $(eV)$ | $R_{\mathrm{SiN}}$ (Å) | $R_{\mathrm{SiH}}$ (Å) | $R_{\mathrm{NH}}$ (Å) |
|---|---|---|---|---|
| Saddle | 1.109601 | 2.778074 | 2.617596 | – |
| Saddle | −3.144738 | 1.523964 | 2.426268 | – |
| Saddle | −5.148745 | 2.006322 | 1.361586 | – |
| Minimum | −6.098598 | 1.529588 | 1.459586 | – |
| Saddle | −5.666608 | 1.575921 | – | 2.969229 |
| Minimum | −9.358509 | 1.523293 | – | 0.998205 |
| Maximum | 1.720515 | 2.647092 | 2.415995 | 3.498876 |
| Saddle | −2.876954 | 1.501907 | 2.309780 | 3.069649 |
| Saddle | −4.908995 | 2.394221 | 2.137984 | 0.974496 |
| Saddle | −0.728138 | 2.155741 | 1.473092 | 2.044809 |
| Saddle | −3.717494 | 1.461352 | 1.634575 | 2.093708 |

*Table 8.* Stationary states for $CS_2$ (Problem 5)

| Type | Energy $(eV)$ | $R_{\mathrm{CS}}$ (Å) | $R_{\mathrm{CS'}}$ (Å) | $R_{\mathrm{SS'}}$ (Å) |
|---|---|---|---|---|
| Saddle | −1.668827 | 2.761779 | – | 2.695109 |
| Saddle | 103.740892 | 0.949956 | – | 1.813411 |
| Minimum | 97.485407 | 0.908824 | – | 1.417728 |
| Minimum | −12.004548 | 1.552422 | 1.552422 | – |
| Saddle | −0.049002 | 4.171034 | 4.171034 | 3.978688 |

*Table 9.* Stationary states for HBO with PES1 (Problem 5)

| Type | Energy $(eV)$ | $R_{\mathrm{BH}}$ (Å) | $R_{\mathrm{BO}}$ (Å) | $R_{\mathrm{OH}}$ (Å) |
|---|---|---|---|---|
| Saddle | −7.598281 | 3.264082 | 1.187662 | – |
| Minimum | −16.678316 | 1.165505 | 1.185028 | – |
| Minimum | −6.556670 | 1.162756 | – | 2.349430 |
| Saddle | −0.216647 | – | 2.554092 | 3.688901 |

## 6. Concluding remarks

We have described here an LP-hard based method to solve the linear interval 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 than the preconditioned interval Gauss–Seidel approach, and thus leads to a

*Table 10.* Stationary states for HBO with PES2 (Problem 5)

| Type | Energy $(eV)$ | $R_{BH}$ (Å) | $R_{BO}$ (Å) | $R_{OH}$ (Å) |
|---|---|---|---|---|
| Minimum | $-16.678851$ | 1.168947 | 1.184167 | – |
| Minimum | $-6.639249$ | 1.154136 | – | 2.344208 |
| Minimum | $-11.305022$ | – | 1.192047 | 2.383483 |
| Saddle | $-11.134196$ | 2.906575 | 1.185979 | 2.398381 |

large reduction in the number of subintervals 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.

We have demonstrated that the interval-Newton approach is a powerful, deterministic approach to the solution of a number of global optimization, as well as nonlinear equation solving problems, such as arise in chemical engineering and other areas of engineering and science. Continuing improvements in methodology, together with advances in software and hardware will make this an increasingly attractive problem solving tool.

## Acknowledgements

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 and Technology Fund.

## References

1. Biegler, L.T. and Tjoa, I.B. (1993), A parallel implementation for parameter estimation with implicit models, *Anns. Opns. Res.*42, 1–23.
2. Esposito, W.R. and Floudas, C.A. (1998), Global optimization in parameter estimiation of nonlinear algebraic models via the error-in-variables approach, *Ind. Eng. Chem. Res.* 37, 1841–1858.
3. Gau, C.-Y. and Stadtherr, M.A. (2000), Reliable nonlinear parameter estimation using interval analysis: error-in-variable approach, *Comput. Chem. Eng.* 24, 631–638.
4. Gau, C.-Y., Brennecke, J.F. and Stadtherr, M.A. (2000), Reliable parameter estimiation in VLE modelling, *Fluid Phase Equilib.* 168, 1–18.
5. Gau , C.-Y. and Stadtherr, M.A. (2002), New interval methodologies for reliable chemical process modelling, *Comput. Chem. Eng.* 26, 827–840.
6. Gau , C.-Y. and Stadtherr, M.A. (2002), Deterministic global optimization for error-in-variables parameter estimation, *AIChE J.* 48, 1191–1197.
7. Hansen, E.R. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.
8. Hua, J.Z., Brennecke, J.F. and Stadtherr, M.A. (1998), Reliable computation of phase stability using interval analysis: cubic equation of state models, *Comput. Chem. Eng.* 22, 1207–1214.
9. Hua, J.Z., Brennecke, J.F. and Stadtherr, M.A. (1998), Enhanced interval analysis for phase stability: cubic equation of state models, *Ind. Eng. Chem. Res.* 37, 1519–1527.

10. Jaulin, L., Kieffer, M., Didrit, O. and Walter, É. (2001), *Applied Interval Analysis*, Springer-Verlag, London.
11. Kearfott, R.B. (1990), Preconditioners for the interval Gauss–Siedel method, *SIAM J. Numer. Anal.* 27, 804–822.
12. Kearfott, R.B. (1991), Decompositon of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems, *Computing* 47, 169–191.
13. Kearfott, R.B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht.
14. Kim, I., Liebman, M. and Edgar, T. (1990), Robust error-in-variables estimation using nonlinear programming techniques, *AIChE J.* 36, 985–993.
15. Maier, R.W., Brennecke, J.F. and Stadtherr, M.A. (1998), Reliable computation of homogeneous azeotropes, *AIChE J.* 44, 1745–1755.
16. Maier, R.W., Brennecke, J.F. and Stadtherr, M.A. (2001), Reliable computation of reactive azeotropes, *Comput. Chem. Eng.* 24, 1851–1858.
17. Maier, R.W. and Stadtherr, M.A. (2001), Reliable density-functional-theory calculations of adsorption in nanoporous materials, *AIChE J.* 47, 1874–1884.
18. Neumaier, A. (1990), *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK.
19. Oettli, W. and Prager, W. (1964), Compatibility of approximated solution of linear equation with given error bounds for coefficients and right-hand sides, *Numerische Mathenatik*, 6, 405–408.
20. Rohn, J. and Kreinovich, V. (1995), Computing exact componentwise bounds on solution of linear systems with interval data is NP-hard, *SIAM J. Matrix. Anal.* 16, 415–420.
21. Schnepper, C.A., Brennecke, J.F. and Stadtherr, M.A. (1995), Robust phase stability analysis using interval methods, *AIChE J. Ser.* 91(304), 356–359.
22. Siirola, J.D., Hauen, S. and Westerberg, A.W. (2002), Agent-based Strategies for Multiobjective Optimization, Paper 265g, *Presented at AIChE Annual Meeting*, Indianapolis, IN, November 3–8.
23. Stradi, B.A., Xu, G., Brennecke, J.F. and Stadtherr, M.A. (2000), Modelling and design of an environmentally benign reaction process, *AIChE Symp. Ser.* 96(323), 371–375.
24. Stradi, B.A., Brennecke, J.F., Kohn, J.P. and Stadtherr, M.A. (2001), Reliable computation of mixture critical points, *AIChE J.* 47, 212–221.
25. Tessier, S.R., Brennecke, J.F. and Stadtherr, M.A. (2001), Reliable phase stability analysis for excess Gibbs energy models, *Chem. Eng. Sci.* 55, 1785–1796.
26. Trefethen, N. (2002), A hundren-dollar hundred-digit challenge, *SIAM News* 35(1), 1.
27. Westerberg, K.M. and Floudas, C.A. (1999), Locating all transition states and studying the reaction pathways of potential energy surfaces, *J. Chem. Phys.* 110, 9259–9295.
28. Xu, G., Scurto, A.M., Castier, M., Brennecke, J.F. and Stadtherr, M.A. (2000), Reliable computation of high pressure solid-fluid equilibrium, *Ind. Eng. Chem. Res.* 39, 1624–1636.
29. Xu, G., Brennecke, J.F. and Stadtherr, M.A. (2002), Reliable computation of phase stability and equilibrium from the SAFT equation of state, *Ind. Eng. Chem. Res.* 41, 938–953.