

Principles of Verified Computations

by

R. Baker Kearfott, rbk@louisiana.edu
*Department of Mathematics, University of
Louisiana at Lafayette*

Many have recently become aware of interval computations, but when such methods are practical is the subject of continuing research. Nonetheless, the ability of interval analysis to computationally verify existence and uniqueness of actual solutions near approximate solutions obtained by traditional methods, as well as to supply mathematically rigorous bounds on these actual solutions for a wide variety of problems, has not been fully appreciated. We will ameliorate that in this talk.

Talk Outline

We will

- introduce the elements of interval arithmetic,
- show how Newton Methods can be used to verify solutions to both linear and nonlinear problems,
- Mention successes in practical areas.

What is Interval Arithmetic?

Interval arithmetic is based on defining the four elementary arithmetic operations on intervals. Let $\mathbf{x} = [\underline{x}, \bar{x}]$ and $\mathbf{y} = [\underline{y}, \bar{y}]$ be intervals. Then, if $op \in \{+, -, *, /\}$, we define

$$\mathbf{x} \text{ op } \mathbf{y} = \{x \text{ op } y \mid x \in \mathbf{x} \text{ and } y \in \mathbf{y}\}.$$

For example, $\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$. In fact, all four operations can be defined in terms of addition, subtraction, multiplication, and division of the endpoints of the intervals, although multiplication and division may require comparison of several results. The result of these operations is an interval except when we compute \mathbf{x}/\mathbf{y} and $0 \in \mathbf{y}$.

Why Interval Arithmetic?

- With *directed rounding*, we can bound roundoff error in the computations. We can combine interval arithmetic with tools such as fixed-point iteration theorems to
 - automatically verify rigorous bounds, from approximate solutions.
 - automatically prove theorems.
- Interval arithmetic provides *rigorous bounds on the ranges* of functions. Possession of such bounds can be powerful, especially in *rigorous global optimization*.

An Example of Interval Arithmetic

$$\begin{aligned}[-1, 2] ([3, 4] + [-5, 6]) &= [-1, 2] ([-2, 10]) \\ &= [-10, 20],\end{aligned}$$

whereas

$$\begin{aligned}[-1, 2][3, 4] + [-1, 2][-5, 6] &= [-4, 8] + [-10, 12] \\ &= [-14, 20]\end{aligned}$$

Here,

$$\begin{aligned}[-14, 20] &= \{x \mid x = x_1 + x_2, x_1 \in [-4, 8], \\ &\quad x_2 \in [-10, 12]\},\end{aligned}$$

and $[-14, 20]$ *contains* the range of $xy + xz$ for $x \in [-1, 2]$, $y \in [3, 4]$, and $z \in [-5, 6]$.

Interval Extensions of Functions

Definition. If f is a continuous function of a real variable, then an interval extension \mathbf{f} is a function from the set of intervals to the set of intervals, such that, if \mathbf{x} is an interval in the domain of \mathbf{f} ,

$$\{f(x) \mid x \in \mathbf{x}\} \subset \mathbf{f}(\mathbf{x}).$$

We also sometimes desire (and it is possible to arrange)

$$\mathbf{x} \subset \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x}) \subset \mathbf{f}(\mathbf{y}).$$

- We may obtain such interval extensions of a polynomial by replacing real operations by corresponding interval operations. For example, if $p(x) = x^2 - 4$, then $\mathbf{p}([1, 2])$ may be defined by

$$\begin{aligned}\mathbf{p}([1, 2]) &= ([1, 2])^2 - 4 = ([1, 4]) - [4, 4] \\ &= [-3, 0].\end{aligned}$$

Interval Extensions of Transcendental Functions

We may use the mean value theorem or Taylor's theorem. For example, say \boldsymbol{x} is an interval and $a \in \boldsymbol{x}$. Then, for any $y \in \boldsymbol{x}$, we have

$$\sin(y) = \sin(a) + (y - a) \cos(a) - \frac{(y - a)^2}{2} \sin(c)$$

for some c between a and y . If a and y are both within a range where the sine function is non-negative, then we obtain

$$\sin(y) \in \sin(a) + (\boldsymbol{x} - a) \cos(a) - \frac{(\boldsymbol{x} - a)^2}{2}.$$

Interval Extensions of Transcendental Functions

(continued)

Specifically, if $\boldsymbol{x} = [.1, .3]$ and we use $a = .2$, we would obtain the “value”

$$\begin{aligned}\mathbf{sin}([.1, .3]) &\subseteq \sin(.2) + [-.1, .1] \cos(.1) \\ &\quad - \frac{([-.1, .1]^2)}{2} \\ &\subseteq [.0998, .0999] \\ &\quad + [-.1, .1][.995, .996] \\ &\quad + [-.005, 0] \\ &= [-.0048, .1995]\end{aligned}$$

- Sharper interval extensions may be obtained in specific cases by using e.g. monotonicity of the original function.

Pitfalls to Naive Interval Arithmetic

- For rigorous bounds on roundoff error, one may be tempted to translate floating-point computer codes by merely changing the data types.
- Due to interval dependencies, such naive development often is unsuccessful.
- Interval arithmetic is successful if it is applied to appropriate tasks and with appropriate algorithms.
 - It provides rigorous results from the computer arithmetic.
 - It can actually result in *significantly faster* algorithms.
- Researchers continue to make advances to enlarge the domain in which interval analysis can make computations rigorous and reliable.

Nonlinear Equations

Interval Newton Methods

Interval analysis can be used either to

- Construct rigorous bounds around an approximate solution, in which an actual solution must lie.
- Exhaustively search a region to find all roots of a nonlinear system.

Verification is easier than exhaustive search. However, both tasks are based on *computational existence / uniqueness theorems*.

Computational Existence / Uniqueness

A Simple Example

Take $n = 1$ and $f(x) = x^2 - 4$, take $\mathbf{x} = [1, 2.5]$, and take an initial guess $\check{x} = 1.75$. Then, following the derivation of the classical Newton method, any solutions x^* of $f(x) = 0$ in \mathbf{x} must obey

$$\begin{aligned} f(x^*) - f(\check{x}) &= f'(\xi)(x^* - \check{x}), \text{ whence} \\ x^* &= \check{x} - \frac{f(\check{x})}{f'(\xi)}, \text{ whence} \\ x^* &\in \check{x} - \frac{f(\check{x})}{\mathbf{f}'(\mathbf{x})}, \end{aligned}$$

where $\mathbf{f}'(\mathbf{x})$ is an interval extension to the derivative $f'(x)$.

$$\tilde{\mathbf{x}} = \check{x} - \frac{f(\check{x})}{\mathbf{f}'(\mathbf{x})}$$

is called an *interval Newton method*.

A Simple Example

(Continued)

In addition, if $\tilde{\mathbf{x}} \subset \mathbf{x}$, connections of interval arithmetic with classical fixed point theory give that there is a unique solution to $f(x) = 0$ in \mathbf{x} , and hence in $\tilde{\mathbf{x}}$.

In particular, for the example, the interval Newton iteration equation becomes

$$\begin{aligned}\tilde{\mathbf{x}} &= \check{x} - \frac{\check{x}^2 - 4}{2\mathbf{x}} \\ &= 1.75 - \frac{-0.9375}{[2, 5]} \\ &= 1.75 - [.1875, .46875] \\ &= [1.9375, 2.21875] \subset \mathbf{x},\end{aligned}$$

so we may conclude that there is a unique root of f in \mathbf{x} .

Computational Existence / Uniqueness

Multivariate Problems

Computational existence / uniqueness works the same way for multivariate problems, with interval extensions to the Jacobi matrix, or, more generally, *Lipschitz matrices* or *slope matrices* replacing the interval derivative. We now outline this in more detail.

Multivariate Existence / Uniqueness

Regularity

Lemma 1 *Suppose $f : \mathbf{x} \rightarrow \mathbb{R}^n$, where*

$$\mathbf{x} = ([\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_n, \bar{x}_n]) \subset \mathbb{R}^n$$

and \mathbf{A} is a Lipschitz matrix, such as a point-wise interval extension of the Jacobian matrix of f over \mathbf{x} . If \mathbf{A} is regular, then any root of f in \mathbf{x} is unique.

Proof: Suppose $x^* \in \mathbf{x}$ and $x \in \mathbf{x}$ have $f(x^*) = 0$ and $f(x) = 0$. If \mathbf{A} is a Lipschitz matrix, then there is an $A \in \mathbf{A}$ such that

$$\begin{aligned} f(x^*) - f(x) &= 0 \\ &= A(x^*) - A(x) \\ &= A(x^* - x). \end{aligned}$$

If $x^* \neq x$, then A would have a null vector, contradicting the regularity of \mathbf{A} . \square

Multivariate Interval Newton Methods

An *interval Newton method* is defined by an iteration of the form

$$\tilde{\mathbf{x}} = \mathbf{N}(f; \mathbf{x}, \check{x}) = \check{x} + \mathbf{v}, \quad (1)$$

where

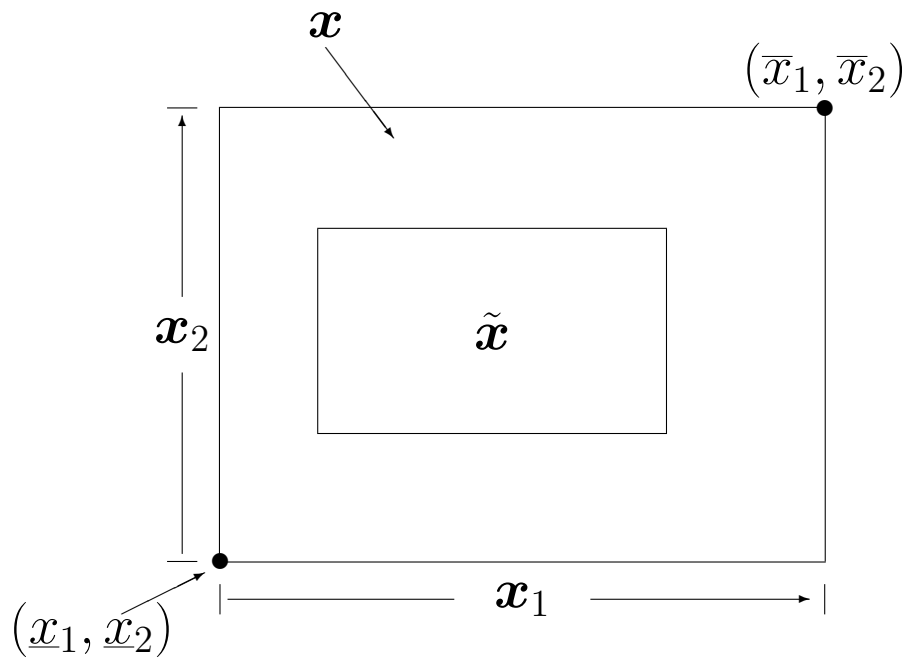
$$\Sigma(\mathbf{A}, -f(\check{x})) \subset \mathbf{v},$$

where \mathbf{A} is a Lipschitz matrix for f over \mathbf{x} and $\Sigma(\mathbf{A}, -f(\check{x}))$ is that set $\{x \in \mathbb{R}^n\}$ such that there exists an $A \in \mathbf{A}$ with $Ax = -f(\check{x})$.

Theorem 2 *Suppose $\tilde{\mathbf{x}}$ is the image of \mathbf{x} under an interval Newton method. If $\tilde{\mathbf{x}} \subseteq \mathbf{x}$, it follows that there exists a unique solution of $f(x) = 0$ within \mathbf{x} .*

Interval Newton Methods

Illustration



In this case, an interval Newton method proves existence.

On Interval Newton Methods

- Proof of the interval Newton existence / uniqueness theorem proceeds from properties of interval arithmetic and the contraction mapping theorem, Miranda's theorem, or the Brouwer fixed point theorem (depending on context and particulars).
- Reasonable bounding sets \mathbf{v} can be obtained by various methods, such as the preconditioned interval Gauss–Seidel method or preconditioned interval Gaussian elimination.
- Iteration of interval Newton methods leads to a locally quadratically convergent process.

A Multivariate Interval Newton Method Example

Interval Gauss–Seidel Method

Assume $f(x) = (f_1(x), \dots, f_n(x))$ is continuously differentiable. Then the mean value theorem gives

$$f_i(x_i) = f_i(x_{-i}, \check{x}_i) + \frac{\partial f_i}{\partial x_i}(x_{-i}, \xi_i)(x_i - \check{x}_i),$$

whence $f_i(x) = 0$ provided

$$\begin{aligned} x_i &= \check{x}_i - f_i(x_{-i}, \check{x}_i) / \frac{\partial f_i}{\partial x_i}(x_{-i}, \xi_i) \\ &\subseteq \check{x}_i - f_i(\mathbf{x}_{-i}, \check{x}_i) / \frac{\partial f_i}{\partial x_i}(\mathbf{x}_{-i}, \mathbf{x}_i) \end{aligned}$$

The above inclusion forms the Interval Gauss–Seidel iteration equation.

Underlying Mathematics to Analyze the Interval Gauss–Seidel Method

Miranda's Theorem

Theorem 3 *Suppose $\mathbf{x} \in \mathbb{IR}^n$, and let the faces of \mathbf{x} be denoted by*

$$\begin{aligned}\mathbf{x}_{\underline{i}} &= (\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \underline{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_n)^T \\ \mathbf{x}_{\bar{i}} &= (\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \bar{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_n)^T.\end{aligned}$$

Let $f = (f_1, \dots, f_n)^T$ be a continuous function defined on \mathbf{x} . If

$$\mathbf{f}_i^u(\mathbf{x}_{\underline{i}}) \mathbf{f}_i^u(\mathbf{x}_{\bar{i}}) \leq 0 \quad (2)$$

for each i between 1 and n , then there is an $x \in \mathbf{x}$ such that $f(x) = 0$.

Interval Gauss–Seidel Method

Verification Properties

Do sequentially for $i = 1$ to n :

1. $\tilde{\mathbf{x}}_i \leftarrow \check{\mathbf{x}}_i - f_i(\mathbf{x}_{-i}, \check{\mathbf{x}}_i) / \frac{\partial f_i}{\partial x_i}(\mathbf{x}_{-i}, \mathbf{x}_i)$

2. $\mathbf{x}_i \leftarrow \tilde{\mathbf{x}}_i$

Then:

1. If $\tilde{\mathbf{x}}_i \subseteq \mathbf{x}_i$ after each step 1, then the hypotheses of Miranda's theorem are satisfied, and, hence, there exists a solution to $f(\mathbf{x}) = 0$ within \mathbf{x} .
2. If $\tilde{\mathbf{x}}_i \subseteq \mathbf{x}_i$ after each step 1, then $\mathbf{f}'(\mathbf{x})$ must be regular, and hence, the solution in \mathbf{x} is also unique.

Note: The system $f(\mathbf{x}) = 0$ is usually *preconditioned* by a point matrix Y to make the Jacobi matrix $Y\mathbf{f}'(\mathbf{x})$ approximately diagonal.

Preconditioned Interval Gauss–Seidel

An Example

$$\begin{aligned}f_1(x_1, x_2) &= x_1^2 - 4x_2, \\f_2(x_1, x_2) &= x_2^2 - 2x_1 + 4x_2,\end{aligned}$$

and

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T = ([-0.1, 0.1], [-0.1, 0.3])^T.$$

$x^* = (0, 0)^T$, of $f(x^*) = 0$ is unique in \mathbf{x} . Take $\check{x} = (0, .1)^T$, so $f(\check{x}) = (-.4, .41)^T$, and

$$\begin{aligned}\mathbf{f}'(\mathbf{x}) &= \begin{pmatrix} 2\mathbf{x}_1 & -4 \\ -2 & 2\mathbf{x}_2 + 4 \end{pmatrix} \\ &= \begin{pmatrix} [-.2, .2] & -4 \\ -2 & [3.8, 4.6] \end{pmatrix}.\end{aligned}$$

An Example

(Continued)

Take Y to be the inverse of the midpoint matrix of $\mathbf{f}'(\mathbf{x})$:

$$Y = \{m(\mathbf{f}'(\mathbf{x}))\}^{-1} = \begin{pmatrix} -.525 & -.5 \\ -.25 & 0 \end{pmatrix}.$$

Then $Y \mathbf{f}'(\mathbf{x}) = \begin{pmatrix} [0.895, 1.105] & [-.2, .2] \\ [-.05, .05] & 1 \end{pmatrix}$,
 $Y f(\tilde{\mathbf{x}}) = (.005, .1)^T$, and

$$\begin{aligned} \tilde{\mathbf{x}}_1 &= 0 - \frac{-.005 + [-.2, .2][-.2, .2]}{[.895, 1.105]} \\ &\subseteq \frac{[-.035, .045]}{[.895, 1.105]} \subseteq [-.0392, .0503] \\ &\subset \mathbf{x}_1 = [-.1, .1]. \end{aligned}$$

Similarly, $\tilde{\mathbf{x}}_2 \subset \mathbf{x}_2$, so there is a unique root of f in \mathbf{x} .

Applications of Interval Newton Existence/Uniqueness

- Global, exhaustive search for all solutions to a nonlinear system of equations.
- Rigorous, tight bounds on solutions to linear and nonlinear systems, given solutions to approximate systems.
- Global optimization.
- Infallible step controls for continuation methods.
- Computation of the topological degree.
- Verification of a particular value of the topological degree.

Finding All Roots

- In *univariate* problems, interval Newton methods can be iterated with *extended interval arithmetic* to *always* find *all* roots.
- In *multivariate* problems, interval Newton methods can be combined with *generalized bisection* and *binary search* to *always* find *all* roots.
- In both instances, failure modes are benign:
 - Failure can only occur by exceeding the computer's resources or because of the limited resolution of the floating-point numbers.
 - When failure occurs, the algorithms still print a list of boxes within which all roots must lie.

Global Optimization

With interval methods, we can:

Find, with certainty, the global minimum of the nonlinear objective function

$$\phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_n) \quad (3)$$

where bounds \underline{x}_i and \bar{x}_i are known with

$$\underline{x}_i \leq x_i \leq \bar{x}_i \text{ for } 1 \leq i \leq n.$$

To do this, a *branch-and-bound* algorithm with the following general features is used.

- A technique for partitioning a region into subregions is combined with a technique for computing a lower bound $\underline{\phi}$ and an upper bound $\bar{\phi}$ of the objective function ϕ over a region \mathbf{x} .
- The subboxes are placed in a list in order of increasing $\underline{\phi}$.
- The list is purged of those boxes for which $\underline{\phi}$ is greater than $\bar{\phi}$ for some other box in the list.
- An interval Newton method accelerates the procedure by quickly and rigorously locating critical points.

Global Optimization

An Example

Suppose we are to minimize

$f(x) = x^2 - 2x + 1$ (written in that way), and the search region is $\mathbf{x} = [.5, 2]$.

- $f((.5 + 2)/2) = f(1.25) = 0.0625$, so the best estimate of the minimum is 0.0625.
- Split $[.5, 2]$ to $[.5, 1.25]$ and $[1.25, 2]$; compute $\mathbf{f}[.5, 1.25] = [-1.25, 1.5625]$; $f(0.875) = 0.015625 < 0.0625$, so 0.015625 is the new best estimate. Store $[.5, 1.25]$ on a list \mathcal{L} .
- $\mathbf{f}([1.25, 2]) = [-1.4375, 2.5]$; $f(1.625) = .390625$, so there is no new best estimate. Store $[1.25, 2]$ on \mathcal{L} before $[.5, 1.25]$, since $-1.4375 < -1.25$.

Optimization

(Continued)

- Continue such processing by popping the first item from the list.
 - if a lower bound on a box is bigger than the best estimate, discard the box.
 - Put a box on a final list if its diameter is small.
- Interval Newton methods can be used to accelerate the process.
- Traditional optimization codes are useful to get good best estimates.

Successes in Practical Areas

Summary

- Linear algebraic systems: A posteriori error bounds (Korn and Ullrich).
- Computer-Aided design and computational geometry (Snyder, Patrikalakis, Maekawa, etc.)
- Chemical engineering models; computational chemistry (Stadtherr and his students)
- Rigorous proofs in computational dynamics (Recent proof by Warwick Tucker that the Lorenz equations have a strange attractor)
- Other mathematical proofs (Thomas Hales' recent proof of the Kepler conjecture)
- Proof of feasibility of particle accelerator design (Martin Berz and his students)
- Proof of bounds on asteroid orbits.

Our Interval-Based Global Optimization Software

GlobSol

- A Fortran 90 package
 - well-tested.
 - self-contained.
- Solves constrained and unconstrained global optimization problems
- Separate program solves square algebraic systems of equations.
- Utility programs for interval and point evaluation, etc.
- Subroutine / module libraries for interval arithmetic, automatic differentiation, etc.
- Publicly available free of charge
http://interval.louisiana.edu/GlobSol/download_GlobSol.html

Use of GlobSol

An Example

The following Fortran 90 program defines the objective function

$$\text{minimize } \phi(X) = -2 * x_1^2 - x_2^2$$

subject to constraints

$$x_1^2 + x_2^2 - 1 \leq 0$$

$$x_1^2 - x_2 \leq 0$$

$$x_1^2 - x_2^2 = 0$$

Use of GlobSol

An Example, continued

```
PROGRAM SIMPLE_MIXED_CONSTRAINTS
  USE CODELIST_CREATION
  PARAMETER (NN = 2)
  TYPE(CDLVAR), DIMENSION(NN) :: X
  TYPE(CDLLHS), DIMENSION(1):: PHI
  TYPE(CDLINEQ), DIMENSION(2) :: G
  TYPE(CDLEQ), DIMENSION(1)    :: C

  CALL INITIALIZE_CODELIST(X)

  PHI(1) = -2*X(1)**2 - X(2)**2
  G(1) = X(1)**2 + X(2)**2 - 1
  G(2) = X(1)**2 - X(2)
  C(1) = X(1)**2 - X(2)**2

  CALL FINISH_CODELIST
END PROGRAM SIMPLE_MIXED_CONSTRAINTS
```

GlobSol Example

(continued)

1. Running the above program produces an internal representation, or code list.
2. The code list is then symbolically differentiated.
3. The optimization code interprets the derivative code list at run time to produce floating point and interval evaluations of the objective function, gradient, and Hessian matrix.
4. A separate data file defines the initial search box, the bound constraints, and the initial guess, if any.
5. A configuration file **GlobSol.CFG** supplies algorithm options, such as which interval Newton method to use and how to precondition the linear systems.

GlobSol Example

The Data File

```
1D-5      ! General domain tolerance
  0  1     ! Bounds on the first variable
  0  1     ! Bounds on the second variable
F F       ! X(1) has no bound constraints F F       ! X
no bound constraints
```

Subsequent optional lines can give an initial guess point.

GlobSol Example

Output File – abridged first part

Output from FIND_GLOBAL_MIN on 04/06/1999 at 08:03:52.
Version for the system is: March 20, 1999

Codelist file name is: MIXEDG.CDL
Box data file name is: MIXED.DT1

Initial box:
[0.0000E+00, 0.1000E+01] [0.0000E+00, 0.1000E+01]

BOUND_CONSTRAINT:
F F F F

CONFIGURATION VALUES:

EPS_DOMAIN: 0.1000E-04 MAXITR: 60000
DO_INTERVAL_NEWTON: T QUADRATIC: T FULL_SPACE: F
VERY_GOOD_INITIAL_GUESS: F
USE_SUBSIT: T
OUTPUT UNIT: 7 PRINT_LENGTH: 1
Default point optimizer was used.

GlobSol Example

Output File – abridged second part

THERE WERE NO BOXES IN COMPLETED_LIST.

LIST OF BOXES CONTAINING VERIFIED FEASIBLE POINTS:

```
Box no.:1
Box coordinates:
 [ 0.7071E+00, 0.7071E+00 ] [ 0.7071E+00, 0.7071E+00 ]
PHI:
 [ -0.1500E+01, -0.1500E+01 ]
Level: 3
Box contains the following approximate root:
 0.7071E+00 0.7071E+00
OBJECTIVE ENCLOSURE AT APPROXIMATE ROOT:
 [ -0.1500E+01, -0.1500E+01 ]
U0:
 [ 0.3852E+00, 0.3852E+00 ]
U:
 [ 0.5777E+00, 0.5777E+00 ] [ 0.0000E+00, 0.1000E+01 ]
V:
 [ 0.1926E+00, 0.1926E+00 ]
INEQ_CERT_FEASIBLE:
 F T
NIN_POSS_BINDING:1

Number of bisections: 1 BEST_ESTIMATE: -0.1500E+01
Total number of boxes processed in loop: 4
Overall CPU time: 0.5000D-01
```