APIC'95, El Paso, Extended Abstracts, A Supplement to the international journal of Reliable Computing

If Input Intervals Are Small Enough, Then Interval Computations Are Almost Always Easy

A. V. Lakeyev[1] and V. Kreinovich[2]

***Abstract*— It is know that many problems in interval computations are NP-hard, including the problem of computing the range $f(\mathbf{x}_1, ..., \mathbf{x}_n)$ of a given polynomial f on given intervals $\mathbf{x}_i$, and a problem of finding the exact interval solution of the system of linear interval equations.**

NP-hard means, crudely speaking, that for every algorithm that produces the exact solution to these problems, there are hard cases for which this algorithm would take exponentially long time to finish its computations. Therefore, if we have a feasible algorithm that finds an interval *enclosure* for the desired solution, this feasible algorithm will inevitably *overestimate* for some (hard) cases.

In this paper, we show that if the input intervals are small enough, then hard cases are *rare* in the following sense: there is a feasible algorithm that computes the precise interval for almost all inputs ("almost all" in a natural sense).

I. The Basic Problem Of Interval Computations Is Known To Be NP-Hard

Definition 1. *By a basic problem of interval computations, we mean the following problem:*

- *Given:*
  - *n intervals $\mathbf{x}_1 = [\tilde{x}_1 - \Delta_1, \tilde{x}_1 + \Delta_1]$, ..., $\mathbf{x}_n = [\tilde{x}_1 - \Delta_1, \tilde{x}_1 + \Delta_1]$, and*
  - *a program that computes a function $f(x_1, ..., x_n)$.*
- *To compute: the range of f, i.e., the endpoints of the interval*

$$\mathbf{y} = f(\mathbf{x}_1, ..., \mathbf{x}_n) =$$
$$\{f(x_1, ..., x_n) \,|\, x_1 \in \mathbf{x}_1, ..., x_n \in \mathbf{x}_n\}.$$

The authors are with [1]Irkutsk Computing Center, Russian Academy of Science, Siberian Division, Lermontov Str. 134, Irkutsk 664033, Russia, email lakeyev@icc.ccsoan.irkutsk.su and with [2]Department of Computer Science, University of Texas at El Paso, El Paso, TX, 79968, email vladik@cs.utep.edu. This work was partially supported by NSF grant No. CDA-9015006 and NASA Research Grant No. NAG 9-757. We are thankful to J. Rohn, S. Shary, and A. Neumaier for the encouragement.

It is known that this problem is *NP-hard* even for polynomial functions f [4].

NP-hard means, crudely speaking, that if we believe that not all problems are easy (the formalization of this natural belief is usually denoted by P$\neq$NP), then for every algorithm that solves this problem, there will be cases on which the running time t grows exponentially with n. So, for some cases of reasonable size (e.g., $n \approx 300$), the running time of this algorithm will exceed the lifetime of the Universe (for precise definitions, see, e.g., [5]).

II. Several Other Problems Of Interval Computations Are Also Known To Be NP-Hard

Several other problems of interval computations are also known to be NP-hard, for example, the problem of solving a system of interval linear equations:

Definition 2.

- *By an interval vector $\mathbf{b}$, we mean a vector $(\mathbf{b}_1, ..., \mathbf{b}_n)$ whose components are intervals.*
- *If $b = (b_1, ..., b_n)$ is a vector, we say that $b \in \mathbf{b}$ if $b_i \in \mathbf{b}_i$ for all i.*
- *By an interval matrix $\mathbf{A}$, we mean a matrix*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \ldots & \mathbf{A}_{1n} \\ & \ldots & \\ \mathbf{A}_{m1} & \ldots & \mathbf{A}_{mn} \end{pmatrix}.$$

- *We say that $A \in \mathbf{A}$ if $A_{ij} \in \mathbf{A}_{ij}$ for all i and j.*

Definition 3. *Assume that we have an interval matrix $\mathbf{A}$ and an interval vector $\mathbf{b}$.*

- *We say that a real-valued vector $x = (x_1, ..., x_n)$ is a possible solution of the system $\mathbf{A}x = \mathbf{b}$, if $Ax = b$ for some $A \in \mathbf{A}$ and $b \in \mathbf{b}$.*
- *We say that x_i is a possible value of $i-$th component x_i of the solution if there exists a possible solution with this very value of x_i.*
- *By an interval of possible values of x_i, we mean the smallest interval that contains all possible values of x_i.*

Definition 4. *By the problem of solving a system of interval linear equations, we mean the following problem:*

- *Given:*
  - *an interval matrix* $\mathbf{A}$*;*
  - *an interval vector* $\mathbf{b}$*;*
  - *an integer* i*;*
- *To compute: the interval of possible values of* x_i.

This problem is also know to be NP-hard, even for the case when we only allow square matrices (with $m = n$) [17].

III. The Usual Interpretation Of These NP-Hardness Results: Every Feasible Enclosure Algorithm Sometimes Overestimates

The usual interpretation of these results is as follows: We have no chance to design a *feasible* algorithm that always finds the exact interval: be it the exact range of the function, or the exact interval of possible values of x_i (in theory of computation, an algorithm is usually described as feasible if its running time is bounded by a *polynomial* $P(n)$ of the length n of the input [9], Section 7.4; [12], Ch. 23). Therefore, we must aim at a feasible algorithm that finds an *enclosure* (i.e., an interval that contains the desired algorithm $\mathbf{y}$).

For such algorithms, NP-hardness can be, crudely speaking, reformulated as follows: if we have a feasible algorithm that always produces an enclosure of the desired interval, then there are cases in which this algorithm will overestimate.

IV. The Existing Interval Enclosure Algorithms Overestimate For Almost All Practical Cases

The existing enclosure methods, starting from naive interval computations [13], do overestimate. But the bad thing is that they overestimate not just in a few cases, but in *almost all* practical cases.

V. We Would Like To Have An Algorithm That Overestimates As Rarely As Possible

Due to NP-hardness results, we cannot hope to improve these methods to the point when they would never overestimate, but it would be nice to have a feasible interval computation algorithm that overestimates as rarely as possible.

VI. Our Main Result: Informal Description

In this paper, such an algorithm is presented for the case when the input intervals are small enough (i.e., when the half-widths Δ_i of these intervals are small enough).

We will show that in this case, we can have an algorithm that almost always (in some reasonable sense) produces the exact interval answer.

VII. Definitions And The Main Result

Definition 5. *Let* $\delta > 0$ *be a real number. We say that an interval* $[\tilde{x} - \Delta, \tilde{x} + \Delta]$ *is* $\delta-$*small if* $\Delta \le \delta$.

Definition 6. *Let the following be given:*

- *a real number* $\varepsilon > 0$*;*
- *a compact domain* $D \subseteq R^n$ *whose Lebesgue measure is positive:* $\mu(D) > 0$ *(for polytopes, Lebesgue measure is a standard volume);*
- *a property* $P(x)$ *(that may be true for some elements* $x \in D$ *and false for some other* $x \in D$*).*

We say that the property P *is true for* $(D, \varepsilon)-$*almost all* x *if* $\mu(\{x \in D \,|\, \neg P(x)\}) \le \varepsilon \cdot \mu(D)$.

Comments.

1. In other words, a property is true for $(D, \varepsilon)-$almost all x if the portion of the domain D for which this property is false does not exceed ε.

2. We want to apply interval estimation algorithms to *computable* functions f, i.e., to the functions that can be computed by a computer program. How to describe such functions?

- Inside the computer, every function f is described as a sequence of arithmetic operations. Everything that we can obtain from the variables $x_1, ..., x_n$ by applying four arithmetic operations is a rational function. So, as a result, what the computer actually computes is always a *rational* function.

  > Even when we write $\sin(x)$, or $\exp(x)$ in a programming language, then, when the compiler compiles this expression, it translates it into a sequence of hardware supported operations, i.e., arithmetic operations. As a result, what actually is computed is a *rational* function that is an *approximation* to the desired one.

- Similarly, when a number is represented in a computer, this number is represented by its finite binary expansion. This expansion is a rational number. So, the rational function f must have rational coefficients.

So, if we are interested in solving the basic problem of interval computations for computable functions, it is sufficient to consider the case of *rational functions* f *with rational coefficients.*

THEOREM 1. *There exists a feasible algorithm $\mathcal{U}$ with the following two properties:*

- *If we are given:*
  - *n intervals $\mathbf{x}_i = [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$, and*
  - *a rational function f with rational coefficients,*

  then this algorithm returns an interval (maybe, an infinite one) that encloses the range $\mathbf{y} = f(\mathbf{x}_1, ..., \mathbf{x_n})$ of the function f.
- *Suppose that we are given:*
  - *a compact domain D of positive Lebesgue measure;*
  - *a rational function f with rational coefficients that is finite in some open neighborhood $N \supseteq D$; and*
  - *a real number $\varepsilon > 0$.*

  Then, there exists a number $\delta > 0$ for which the following is true:

  For $(D, \varepsilon)-$almost all $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_n)$, if all n intervals $\mathbf{x}_i = [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$ are $\delta-$small, then this algorithm $\mathcal{U}$ returns the exact value of the range $f(\mathbf{x}_1, ..., \mathbf{x}_n)$.

THEOREM 2. *There exists a feasible algorithm $\mathcal{U}$ with the following two properties:*

- *If we are given:*
  - *a square $n \times n$ interval matrix $\mathbf{A}$ with elements $\mathbf{A}_{ij} = [\tilde{A}_{ij} - \Delta_{ij}, \tilde{A}_{ij} + \Delta_{ij}]$;*
  - *an $n-$dimensional interval vector $\mathbf{b}$ with elements $\mathbf{b}_i = [\tilde{b}_i - \Delta_i, \tilde{b}_i + \Delta_i]$, and*
  - *an integer $i \leq n$;*

  then this algorithm returns an interval (maybe, an infinite one) that encloses the interval of possible values of x_i.
- *Suppose that we are given:*
  - *a compact domain D of positive $(n \times n) + n-$dimensional Lebesgue measure, and an open neighborhood $N \supseteq D$ such that if $(A, b) \in N$, then the matrix A is non-degenerate (i.e., invertible).*
  - *a positive real number $\varepsilon > 0$.*

  Then, there exists a number $\delta > 0$ for which the following is true:

  For $(D, \varepsilon)-$almost all pairs $(\tilde{A}, \tilde{b}) = (\tilde{A}_{11}, \tilde{A}_{12}, ..., \tilde{A}_{nn}, \tilde{b}_1, ..., \tilde{b}_n)$, if all intervals $\mathbf{A}_{ij} = [\tilde{A}_{ij} - \Delta_{ij}, \tilde{A}_{ij} + \Delta_{ij}]$, and $\mathbf{b}_i = [\tilde{b}_i - \Delta_i, \tilde{b}_i + \Delta_i]$ are $\delta-$small, then this algorithm $\mathcal{U}$ returns the exact interval of possible values of x_i.

Comment. In other words, if the intervals are small enough, then these algorithms give *exact* intervals in almost all cases.

VIII. Proofs

A. Proof of Theorem 1

A.1. A comment on the proof

We will prove that this theorem is true for a quite reasonable algorithm $\mathcal{U}$ (that we will describe below). This algorithm has, in effect, been used many times before, and it is not our main achievement. Our main achievement is the observation that this algorithm gives an exact estimate in almost all cases.

A.2. Description of the algorithm. Stage 1: simplifying f

First, we must check whether the given function f actually depends on all of its variables.

To do that, we will:

- apply analytical differentiation formulas to f (see, e.g., [15]), and
- check whether the resulting derivatives

$$f_{,i} = \frac{\partial f}{\partial x_i}$$

  are identically 0 or not.

The derivative of a rational function is also rational. Therefore, this rational function can be represented as a ratio of two polynomials $f_{,i} = P/Q$. This ratio is identically equal to 0 iff the polynomial P is identically equal to 0, and the polynomial is identically 0 iff all its coefficients are zeros. Since all the coefficients of f are rational numbers, the coefficients of $f_{,i}$ (and thus, of P) are also rational numbers, and for rational numbers, equality to 0 is easy to check.

If for some i, it turns out that the partial derivative $f_{,i}$ is identically 0, then f does not depend on this variable x_i at all. So, we can substitute an arbitrary value of x_i into the original formula f, and get a new expression with one fewer variable.

We can repeat this procedure for all the variables x_i for which $f_{,i}$ is identically 0. As a result, we get a new expression for f that only contains the variables on which f essentially depend (i.e.g, for which the derivative $f_{,i}$ is not identically 0).

Comment. To explain why this procedure is so complicated, let us give an example of a rational function f that does not depend on x_3 but whose expression explicitly contains x_3:

$$f(x_1, x_2, x_3) = \frac{x_1^2 + x_1 x_2 + x_1 x_3 + x_2 x_3}{x_1^2 + x_1 x_3}.$$

If we factorize the numerator and the denominator, we will see that this function f, indeed, does not depend on x_3:

$$f = \frac{(x_1 + x_2)(x_1 + x_3)}{x_1(x_1 + x_3)} = \frac{x_1 + x_2}{x_1}.$$

We do not want to factorize, because factorization can be a very time-consuming procedure. Instead, we differentiate with respect to all three variables, and indeed get $f_{,3} = 0$. As a result, we substitute an arbitrary constant instead of x_3, and get an equivalent expression for f that contains only two variables. For example, if we substitute $x_3 = 0$, we get the following expression:

$$f(x_1, x_2) = \frac{x_1^2 + x_1 x_2}{x_1^2}.$$

A.3. Description of the algorithm. Stage 2

After stage 1, we have a rational expression $f(x_1, ..., x_m)$ for which for all $i \le m$, the derivative $f_{,i}$ is not identically 0. For this expression, we do the following:

- First, for each i from 1 to m, we apply an interval centered form [18] to compute the interval enclosure $\mathbf{e}_i = [e_i^-, e_i^+]$ for $f_{,i}(\mathbf{x}_1, ..., \mathbf{x}_m)$.
- If for some i, the interval $\mathbf{e}_i$ contains 0, then we apply the same interval center form (or any other interval computation technique) to compute an enclosure $\mathbf{e}$ for the desired range $f(\mathbf{x}_1, ..., \mathbf{x}_m)$.
- If for all i, the interval $\mathbf{e}_i$ does not contain 0, then we compute $s_i = \text{sign}(e_i^+)$ (where $\text{sign}(a) = 1$ for $a > 0$ and $\text{sign}(a) = -1$ if $a < 0$), and return the interval $[y^-, y^+]$, where

  $$y^- = f(\tilde{x}_1 - s_1\Delta_1, ..., \tilde{x}_m - s_m\Delta_m),$$

  $$y^+ = f(\tilde{x}_1 + s_1\Delta_1, ..., \tilde{x}_m + s_m\Delta_m),$$

  as the exact value of the range.

A.4. Proof that this algorithm always computes the enclosure

If $0 \in \mathbf{e}_i$, then the fact that the above-described algorithm returns an enclosure follows from the properties of interval computation methods.

If $0 \notin \mathbf{e}_i$ for all i, this means that for every i, this means that for $(x_1, ..., x_m) \in \mathbf{x}_1 \times ... \times \mathbf{x}_m$, the function f is *monotonic* with respect to each of its variables:

- If $\mathbf{e}_i > 0$, then $f_{,i} > 0$, and f is *increasing* w.r.t. x_i.
- If $\mathbf{e}_i < 0$, then $f_{,i} < 0$, and f is *decreasing* w.r.t. x_i.

If a function f is increasing w.r.t. x_i, then its maximum is attained when x_i takes the largest possible values. On the interval $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$, the largest possible value is $x_i = \tilde{x}_i + \Delta_i$.

If a function f is decreasing w.r.t. x_i, then its maximum is attained when x_i takes the smallest possible values. On the interval $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$, the smallest possible value is $x_i = \tilde{x}_i - \Delta_i$.

We can combine these two cases by saying that if f is monotone w.r.t. x_i, then the maximum of f is attained for $x_i = \tilde{x}_i + s_i\Delta_i$.

In our case, f is monotone w.r.t. all its variables, so, its maximum is attained when each of its variables x_i takes the value $x_i = \tilde{x}_i + s_i\Delta_i$. This maximum is therefore equal to

$$f(\tilde{x}_1 + s_1\Delta_1, ..., \tilde{x}_m + s_m\Delta_m).$$

Similarly, one can prove that the minimum of the set of possible values of f (i.e., the lower endpoint of the range interval $f(\mathbf{x}_1, ..., \mathbf{x}_m)$ is equal to

$$f(\tilde{x}_1 - s_1\Delta_1, ..., \tilde{x}_m - s_m\Delta_m).$$

So, in this case, the result of our algorithm not only *encloses* the desired range, it *coincides* with this range.

Comments.

1. As we have already mentioned, the use of monotonicity is not a novel idea. As examples of successful applications of this idea, we can cite [1, 2, 8, 20, 6, 14, 19, 16, 10, 3, 11].

2. To complete our proof, we must show that the described algorithm returns the exact range in almost all cases.

A.5. Proof that this algorithm almost always computes the exact range

After the first stage, we have a rational function $f(x_1, ..., x_m)$ that depends on each of its m variables. This rational function is finite on some open neighborhood N of D. Since N is an open neighborhood, there exists a $\delta_0 > 0$ such that every point that is $\le \delta_0-$close to the set D (in the sup metric $d(x, y) = \max |x_i - y_i|$) also belongs to N. Let us denote the set of all the points that are δ_0-close to D (i.e., a δ_0-neighborhood of D) by D_0. This set D_0 is closed and bounded (since D is bounded), so, D_0 is a compact.

A rational function f is continuous in every point x in which it is finite. Since f is finite on all points of N, it is also finite on all points from $D_0 \subseteq N$. Therefore, f is continuous on every point of D_0.

Since f is rational, for every i, the partial derivative $f_{,i}$ is also a rational function, i.e., a ratio of two polynomials: $f_{,i} = P_i/Q_i$. Therefore, the set S_i of all points $x \in D$ for which $f_{,i}(x) = 0$ is a set of solutions of the equation $P_i(x) = 0$ for some polynomial P_i. From the topological viewpoint, the structure of the set of all the roots of a polynomial (that is not identically 0) is well known: it is either a manifold of dimension $\le m-1$, or a union of finitely many manifolds of dimension $\le m-1$. In both cases, the Lebesgue measure $\mu(S_i)$ of this set is equal to 0:

$$\mu(S_i) = \mu(\{x \mid f_{,i}(x) = 0\}) = 0.$$

In our algorithm, we do not use the exact values of $f_{i,}$, we only use an *estimate* (i.e., in general, an *approximation*) to the range of $f_{,i}$. Therefore, to analyze the behavior of our algorithm, we would like to have some measure estimate that is based not on the exact value of $f_{,i}(x)$, but on the interval of possible values. Such an estimate is easy to get: Indeed, S_i is an intersection of the sequence of monotone decreasing sequence of sets:

$$S_i = \{x \mid f_{,i}(x) = 0\} = \bigcap_{k=1}^{\infty} \{x \mid |f_{,i}(x)| \le 2^{-k}\}.$$

Therefore,

$$\mu(\{x \mid |f_{,i}(x)| \le 2^{-k}\}) \to \mu(S_i) = 0$$

as $k \to \infty$. So, there exists a k_i for which

$$\mu(\{x \mid |f_{,i}(x)| \le 2^{-k_i}\}) \le \frac{\varepsilon \cdot \mu(D)}{m}.$$

If we denote by k the largest of these k_i:

$$k = \max(k_1, ..., k_m),$$

then we will conclude that

$$\mu(\{x \mid |f_{,i}(x)| \le 2^{-k}\}) \le$$

$$\mu(\{x \mid |f_{,i}(x)| \le 2^{-k_i}\}) \le \frac{\varepsilon \cdot \mu(D)}{m}.$$

Let us denote the corresponding set

$$\{x \mid |f_{,i}(x)| \le 2^{-k}\}$$

by A_i. Then, for a union A of these sets

$$A = \bigcup_{i=1}^{m} A_i = \{x \mid |f_{,i}(x)| \le 2^{-k} \text{ for some } i\},$$

we have

$$\mu(A) \le \sum_{i=1}^{m} \mu(A_i) \le \sum_{i=1}^{m} \frac{\varepsilon \cdot \mu(D)}{m} = \varepsilon \cdot \mu(D).$$

We want to find δ for which the above-described algorithm computes the precise range for all $\delta-$small intervals whose centers $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_n)$ satisfy the condition $\tilde{x} \notin A$. Let us find such a δ. For that, we will use the following two ideas:

- For every i from 1 to m, the rational function $f_{,i}$ is continuous on a compact D_0. Therefore, $f_{,i}$ is uniformly continuous. In particular, there exists a $\delta_i > 0$ such that if $d(x, \tilde{x}) \le \delta_i$, then $|f_{,i}(x) - f_{,i}(\tilde{x})| \le 2^{-(k+2)}$.
- According to [18], the centered method has an accuracy $\le O(\max(\Delta_1, ..., \Delta_m))$. This means that for every i from 1 to m, there exists a constant C_i such that if all input intervals are $\delta-$small, then the difference between the endpoints of the actual range $[a_i^-, a_i^+] = f_{,i}(\mathbf{x}_1, ..., \mathbf{x}_m)$ and the corresponding endpoints of the estimated range $\mathbf{e}_i = [e_i^-, e_i^+]$ does not exceed $C_i \cdot \delta$:

$$|e_i^- - a_i^-| \le C_i \cdot \delta, \;\; |e_i^+ - a_i^+| \le C_i \cdot \delta.$$

Now, let us take

$$\delta = \min(\delta_0, \delta_1, ..., \delta_m, \frac{2^{-(k+2)}}{C_1}, ..., \frac{2^{-(k+2)}}{C_1}).$$

Let us show that for this δ, if the input intervals are $\delta-$small, and $\tilde{x} \notin A$, then none of the intervals $\mathbf{e}_i$ contain zero and therefore, for these algorithms, the above-described algorithm returns the exact interval range.

Indeed, since $\tilde{x} \notin A$, by definition of the set A, we have $|f_{,i}(\tilde{x})| > 2^{-k}$ for all i. This means that either $f_{,i}(\tilde{x}) > 2^{-k}$, or $f_{,i}(\tilde{x}) < -2^{-k}$.

- Let us first consider the first case. In this case, if

  $$x_j \in \mathbf{x}_j = [\tilde{x}_j - \Delta_j, \tilde{x}_j + \Delta_j],$$

  then (since the interval $\mathbf{x}_j$ is $\delta-$small), we have

  $$|x_j - \tilde{x}_j| \le \Delta_j \le \delta.$$

  Hence,

  $$d(x, \tilde{x}) = \max |x_j - \tilde{x}_j| \le \delta.$$

  Due to our choice of δ as $\min(..., \delta_i, ...)$, we have $\delta \le \delta_i$ and therefore, $d(x, \tilde{x}) \le \delta_i$. Due to our choice of δ_i this inequality, in turn, leads to the inequality $|f_{,i}(x) - f_{,i}(\tilde{x})| \le 2^{-(k+2)}$. In particular, we have $f_{,i}(x) \ge f_{,i}(\tilde{x}) - 2^{-(k+2)}$. Since $f_{,i}(\tilde{x}) > 2^{-k}$, we can conclude that

  $$f_{,i}(x) > 2^{-k} - 2^{-(k+2)} = 3 \cdot 2^{-(k+2)}.$$

  So, if $x_1 \in \mathbf{x}_1$, ..., and $x_m \in \mathbf{x}_m$, then $f_{,i}(x) \ge 3 \cdot 2^{-(k+2)}$. Therefore, the lower bound a_i^- of the actual range $f_{,i}(\mathbf{x}_1, ..., \mathbf{x}_m)$ satisfies the inequality

  $$a_i^- \ge 3 \cdot 2^{-(k+2)}.$$

Now, because of our choice of δ as

$$\delta = \min(..., 2^{-(k+2)}/C_i, ...),$$

we have $\delta \le 2^{-(k+2)}/C_i$. Hence,

$$C_i \cdot \delta \le 2^{-(k+2)}.$$

By the choice of C_i, this means that $|e_i^- - a_i^-| \le 2^{-(k+2)}$. Hence, $e_i^- \ge a_i^- - 2^{-(k+2)}$. We already know that $a_i^- \ge 3 \cdot 2^{-(k+2)}$. Hence,

$$e_i^- \ge 3 \cdot 2^{-(k+2)} - 2^{-(k+2)} = 2 \cdot 2^{-(k+2)} > 0.$$

The lower bound of the interval $\mathbf{e}_i$ is positive, hence, this interval can only contain positive numbers and cannot contain 0.

- If $f_{,i}(\tilde{x}) < -2^{-k}$, then we can similarly conclude that the upper bound e_i^+ of the interval $\mathbf{e}_i$ is negative; hence, this interval contains only negative numbers and cannot contain 0.

So, for these intervals, the above algorithm gives the exact range. Q.E.D.

B. Proof of Theorem 2

Theorem 2 follows from Theorem 1 if we take into consideration the fact that if A is invertible, then the solution $x = A^{-1}b$ of the system of linear equations $Ax = b$ is a rational function of the coefficients A_{ij} and b_i.

References

[1] L. Collatz, "Aufgaben monotoner Art", *Arch. Math.*, 1952, Vol. 3, pp. 366–376.

[2] L. Collatz, **The Numerical Treatment of Differential Equations**, Springer-Verlag, 1960.

[3] N. Dimitrova and S. Markov, "On the interval-arithmetic presentation of the range of a class of monotone functions of many variables", In: E. Kaucher, S. M. Markov, and G. Mayer (eds.), **Computer Arithmetic, Scientific Computation and Mathematical Modelling**, J. C. Baltzer Co., IMACS, 1991, pp. 213–228.

[4] A. A. Gaganov, "Computational complexity of the range of the polynomial in several variables", *Cybernetics*, 1985, pp. 418–421.

[5] M. R. Garey, D. S. Johnson, D. S. **Computers and intractability: a guide to the theory of NP-completeness**, W. F. Freeman, San Francisco, 1979.

[6] G. W. Harrison, "Dynamic models with uncertain parameters", In: X. J. R. Avul (Editor), *Proceedings of the First International Conference on Mathematical Modeling, St. Louis*, University of Missouri-Rolla, 1977, pp. 295–304.

[7] V. Kreinovich, A. V. Lakeyev, and S. I. Noskov. "Optimal solution of interval linear systems is intractable (NP-hard)." *Interval Computations,* 1993, No. 1, pp. 6–14.

[8] V. Lakshmikantham, S. Leela, **Differential and integral inequalities**, Academic Press, N.Y., 1969.

[9] H. R. Lewis, C. H. Papadimitriou, **Elements of the Theory of Computations**, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[10] R. Mannshardt, "Enclosing a solution of an ordinary differential equation by sub- and superfunctions", In: C. Ulllrich (ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical methods*, J. C. Baltzer AG, IMACS, 1990, pp. 319–328.

[11] S. M. Markov, "On the presentation of ranges of monotone functions using interval arithmetic", *Proceedings of the International Conference on Interval and Stochastic Methods in Science and Engineering INTERVAL'92*, Moscow, 1992, Vol. 2, pp. 66–74.

[12] J. C. Martin. **Introduction to Languages and the Theory of Computation**, McGraw-Hill, N.Y., 1991.

[13] R. E. Moore, **Interval analysis**, Prentice Hall, Englewood Cliffs, NJ, 1966.

[14] R. E. Moore, **Methods and applications of interval analysis**, SIAM, Philadelphia, 1979.

[15] L. B. Rall, **Automatic differentiation: techniques and applications**, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1981, Vol. 120.

[16] L. B. Rall, "Mean value and Taylor forms in interval analysis", *SIAM J. Math. Anal.*, 1983, Vol. 2, pp. 223–238.

[17] J. Rohn, V. Kreinovich, "Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard", *SIAM Journal on Matrix Analysis and Applications (SIMAX),* 1995, Vol. 16, No. 2 (to appear).

[18] H. Ratschek and J. Rokne, "Optimality of the Centered Form", In: K. Nickel (ed.), **Interval Mathematics 1980,** Acad. Press, N.Y., 1980, pp. 499–508.

[19] J. Schröder, **Operator Inequalities**, Academic Press, N.Y., 1980.

[20] W. Walter, **Differential and integral inequalities**, Springer-Verlag, N.Y., 1970.