Reliable Computing 2 (4) (1996), pp. 341–350

Linear interval equations: Computing enclosures with bounded relative or absolute overestimation is NP-hard

VLADIK KREINOVICH and ANATOLY V. LAKEYEV

It is proved that for every $\delta > 0$, if there exists a polynomial-time algorithm for enclosing solutions of linear interval equations with relative (or absolute) overestimation better than δ, then P = NP. The result holds for the symmetric case as well.

Интервальные системы линейных уравнений: вычисление интервальных оценок с ограниченной относительной или абсолютной погрешностью является NP-трудным

В. КРЕЙНОВИЧ, А. ЛАКЕЕВ

Доказано, что для любого $\delta > 0$, если существует алгоритм с полиномиальным временем выполнения для локализации решений интервальной системы линейных уравнений с относительной (или абсолютной) погрешностью, меньшей δ, то P = NP. Результат справедлив также для случая симметричных систем.

1. Introduction

For a system of linear interval equations

$$A^I x = b^I \tag{1}$$

where A^I is an *interval matrix* (i.e., matrix with interval components), and b^I is an *interval vector* (i.e., vector with interval components), a *solution set* is defined as follows:

$$X = \{x;\ Ax = b \text{ for some } A \in A^I,\ b \in b^I\}.$$

Ideally, for a given linear interval equation, we would like to know the exact bounds of possible values of x_i, i.e., the interval vector $[\underline{x}, \overline{x}]$ given by

$$\underline{x}_i = \min_X x_i, \quad \text{and} \quad \overline{x}_i = \max_X x_i.$$

This work was partially supported by NSF Grants No. CDA–9015006 and EEC–9322370, and by NASA Grant No. NAG 9–757. The authors are greatly thankful to Baker Kearfott, Arnold Neumaier, and especially, to Jiri Rohn. Jiri's contribution has actually made this paper possible to an extent that we feel him practically as its co-author.

In [5], it is proven that the problem of computing the exact bounds is NP-hard (computationally intractable).

Comment. Crudely speaking, NP-hardness of a problem P means that if we are able to solve this problem in reasonable time, then we would be able to solve all problems from a very large class of complicated problems (called class NP) in reasonable time, and this possibility is widely believed to be impossible. Here, by a *reasonable time*, we mean a time that does not exceed some polynomial of the length of the input. For exact definitions see, e.g., [3].

The result from [5] was proven for rectangular (non-square) matrices. In [11], it was shown that even if we restrict ourselves to quadratic interval matrices A^I, computing the *exact* bounds $\underline{x}_i$ and $\overline{x}_i$ is still NP-hard. So, if $P \neq NP$, no feasible (polynomial time) algorithm can compute the *exact* bounds.

These results do not mean that solving linear interval equations is a hopeless task. There exist many efficient algorithms that produce good approximations to the desired bounds; these algorithms can be found, e.g., in Alefeld and Herzberger [2], and in Neumaier [8]. These algorithms do not always produce the exact bounds, but it has been proven [7] that if the interval components of A^I and b^I are "thin" enough, then there exists a polynomial-time algorithm that computes the exact bounds for X in "almost all" cases ("almost all" in some reasonable sense).

Since we cannot always compute the *exact* bounds, the natural question is: would it be possible to have a feasible algorithm if we only want to compute *approximations* to the bounds of X?

In [6], it is shown that for each $\delta > 0$, if we want to compute the bounds that are δ-accurate (i.e., estimates that differ by $\leq \delta$ from the actual bounds) then the problem is also NP-hard. This result is proved for generic rectangular matrices.

J. Rohn [9, 10] has shown that for square matrices, computing approximate bounds is also NP-hard. To formulate his result, we will need the following definition:

Definition 1.

1) *For a system of linear interval equations (1), enclosure is defined as an interval vector $[\underline{y}, \overline{y}]$ satisfying $X \subseteq [\underline{y}, \overline{y}]$, where X is the solution set of (1).*
2) *An interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ is called strongly regular if $\rho(|A_c^{-1}|\Delta) < 1$ (where ρ denotes a spectral radius of a matrix).*

Comment. The condition of strong regularity is known to guarantee that every matrix $A \in A^I$ is regular.

Theorem (Rohn [10]). *Suppose there exists a polynomial-time algorithm which for each strongly regular $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X satisfying*

$$\left|\frac{\overline{y}_i - \overline{x}_i}{\overline{x}_i}\right| \leq \frac{4}{n^2} \tag{2}$$

for each i with $\overline{x}_i \neq 0$. Then $P = NP$.

This theorem shows that computing "sufficiently accurate enclosures" is NP-hard, i.e., if P≠NP, then every algorithm that computes sufficiently accurate estimates requires lots of computation time. Rohn's result is based on the assumption that the larger n, the more accurately we want to compute the enclosures. The natural next question is: what if we want an algorithm to compute all enclosures with the same accuracy? Will it still be an NP-hard problem? In other words, for a given $\delta > 0$, is the problem of computing δ-accurate enclosure for solutions of interval linear systems with square A^I NP-hard? This problem was first formulated by A. Neumaier, whose hypothesis was that this problem was NP-hard.

In this paper, we prove Neumaier's hypothesis (Theorems 1 and 2). We also prove that a similar result is true for the symmetric case (Theorems 3 and 4).

2. The main results

Theorem 1. *Suppose for some real number $\delta > 0$, there exists a polynomial-time algorithm which for each strongly regular $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X satisfying*

$$\left|\frac{\overline{y}_i - \overline{x}_i}{\overline{x}_i}\right| \leq \delta \tag{3}$$

for each i with $\overline{x}_i \neq 0$. Then P = NP.

Theorem 2. *Suppose for some real number $\delta > 0$, there exists a polynomial-time algorithm which for each strongly regular $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X satisfying $|\overline{y}_i - \overline{x}_i| \leq \delta$ for all i. Then* P = NP.

Comments.

1) Hence, the problem of computing sufficiently accurate enclosures is very difficult: an existence of a polynomial-time algorithm yielding the accuracy (3) would imply polynomial-time solvability of all problems in the class NP. As we have already mentioned, this possibility is considered highly unlikely.

2) If P $\neq$ NP, then for absolute accuracy, not only we cannot compute enclosures with one and the same accuracy (i.e., with one and the same bound for absolute overestimation) for all n in reasonable time, but even if we allow accuracy to decrease polynomially with n, we still will not be able to compute these "relaxed-accuracy" enclosures:

Theorem 2′. *Suppose for some polynomial $\delta(n)$, there exists a polynomial-time algorithm which for each strongly regular $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X satisfying $|\overline{y}_i - \overline{x}_i| \leq \delta(n)$ for all i. Then* P = NP.

Proofs: Main Lemma. Our proofs of Theorems 1, 2, and 2′ will use the proof from [10]. In [10], it was proven not only that the problem of computing $(4/n^2)$-accurate enclosures is NP-hard for arbitrary square matrices A^I, but that this problem is NP-hard even if we restrict ourselves to the following special class of linear systems. Let us fix an integer p, and denote $e = (1, 1, \ldots, 1)^T \in R^p$. We will use the matrix norm

$$\|M\|_s = e^T|M|e = \sum_i \sum_j |m_{ij}|.$$

A real symmetric $p \times p$ matrix $M = (m_{ij})$ is called an *MC-matrix* if it is of the form

$$m_{ij} \begin{cases} = p & \text{if } i = j, \\ \in \{0, -1\} & \text{if } i \neq j \end{cases}$$

$(i, j = 1, \dots, p)$. For a given $p \times p$ MC-matrix M, Rohn [10] considers the linear interval system (1) with $A^I = [A_c - \Delta, A_c + \Delta]$, $b^I = [b_c - \delta, b_c + \delta]$ given by

$$A_c = \begin{pmatrix} 0 & -I & 0 \\ -I & 0 & M^{-1} \\ 0 & M^{-1} & M^{-1} \end{pmatrix}, \qquad \Delta = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta ee^T \end{pmatrix} \tag{4}$$

(all the blocks are $p \times p$, I is the unit matrix),

$$b_c = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \delta = \begin{pmatrix} 0 \\ 0 \\ \beta e \end{pmatrix} \tag{5}$$

(all the blocks are $p \times 1$) and

$$\beta = \frac{1}{\|M\|_s + 1}. \tag{6}$$

For each MC-matrix M, the matrix A^I is strongly regular, and the problem of computing the $(4/n^2)$-accurate bounds for the resulting system $A^I x = b^I$ is NP-hard.

In [10], it is also shown that for each system of this type, we have $\overline{x}_1 = \overline{x}_2 = \cdots = \overline{x}_p \geq 1/(2p^2)$, and that the vector $\overline{x}$ is achieved as the solution of a system $Ax = b$, where $b \in b^I$ and

$$A = \begin{pmatrix} 0 & -I & 0 \\ -I & 0 & M^{-1} \\ 0 & M^{-1} & M^{-1} - \beta zz^T \end{pmatrix} \in A^I.$$

We will use these results from [10] as the Main Lemma for our Theorems.

Proof of Theorem 1. Suppose that for some real number $\delta > 0$, there exists a polynomial-time algorithm $\mathcal{U}$ that for each strongly regular $n \times n$ interval matrix A^I and each interval vector b^I computes an enclosure that satisfies the inequality $|\overline{y}_i - \overline{x}_i|/\overline{x}_i \leq \delta$ for all i for which $\overline{x}_i \neq 0$.

To prove that P = NP, we will only need this algorithm applied to systems $A^I x = b^I$ described in the lemma, and to the following simple modifications of these systems: To describe these modifications, we must consider the following auxiliary vector

$$v = \begin{pmatrix} e \\ 0 \\ 0 \end{pmatrix}.$$

For the interval matrix A^I (from (4)), the only difference between $3p \times 3p$ matrices $A \in A^I$ and A_c can be in the right lower $p \times p$ part (because this is where Δ has non-zero elements). Since our vector v has 0 values for its last p elements, we can conclude that for every $A \in A^I$, we have $Av = A_c v$. Therefore, for every real number μ, if $Ax = b$ for some $A \in A^I$ and $b \in b^I$, then $A(x - \mu \cdot v) = Ax - \mu \cdot (Av) = b - \mu \cdot A_c v$. In other words, if x belongs to the solution set X of the original interval system, then $\tilde{x} = x - \mu \cdot v$ belongs to the solution set $\tilde{X}$

of the auxiliary system $A^I\tilde{x} = \tilde{b}^I$, where $\tilde{b}^I = b^I - \mu \cdot A_c v$. This auxiliary system is the desired modification.

Vice versa, if $\tilde{x}$ belongs to $\tilde{X}$, this means that $A\tilde{x} = b$ for some $A \in A^I$ and $\tilde{b} \in \tilde{b}^I$. Then, for $x = \tilde{x} + \mu \cdot v$, we have $Ax = A\tilde{x} + \mu \cdot Av = \tilde{b} + \mu \cdot A_c v$ and therefore, $Ax \in b^I = \tilde{b}^I + \mu \cdot A_c v$. So, if $\tilde{x} \in \tilde{X}$, then $x = \tilde{x} + \mu \cdot v \in X$.

Therefore, a vector x belongs to the solution set X of the original system iff the vector $\tilde{x} = x - \mu \cdot v$ belongs to the solution set of the auxiliary system. Hence, the optimal enclosure $\bar{\tilde{x}}$ for the new system is related to the optimal enclosure $\bar{x}$ for the original system $A^I x = b^I$ by a simple formula:

$$\bar{\tilde{x}}_i = \bar{x}_i - \mu, \quad i = 1, \ldots, p. \tag{7}$$

We will show that for interval systems described in the lemma and for their above-described modifications, by applying the given algorithm $\mathcal{U}$ several times (to different auxiliary interval systems), we can design new algorithms $\mathcal{U}(k)$ that compute enclosures $\bar{y}(k)$, $k = 1, 2, \ldots$ for these systems, and for which $|\bar{y}_i(k) - \bar{x}_i|/\bar{x}_i \leq \delta_k$ with decreasing δ_k. As a result (as we will show), after no more than a polynomial number of applications of $\mathcal{U}$, we will get an enclosure with a relative accuracy $4/n^2$ (the value of parameter k that corresponds to this accuracy will depend on the size n of the system). The number of applications of $\mathcal{U}$ does not exceed a polynomial of n, and each application requires a computation time bounded by a polynomial of n. Therefore, all computations that result in a $(4/n^2)$-accurate enclosure, are performed in polynomial time. Hence, from the lemma, we will conclude that P = NP.

Base. First, we apply $\mathcal{U}$ to such systems, and get an enclosure $[\underline{y}, \bar{y}]$ for which $|\bar{y}_i - \bar{x}_i|/\bar{x}_i \leq \delta$. So, the first algorithm $\mathcal{U}(1)$ is simply $\mathcal{U}$, and we have the first enclosure $\bar{y}(1) = \bar{y}$ with $\delta_1 = \delta$.

Iteration step. Suppose that we have an algorithm $\mathcal{U}(k)$ that for systems from the lemma and for the above-described auxiliary systems, computes an enclosure $\bar{y}(k)$ satisfying the inequality

$$\left|\frac{\bar{y}_i(k) - \bar{x}_i}{\bar{x}_i}\right| \leq \delta_k$$

for all $i \leq p$ for which $\bar{x}_i \neq 0$. In other words, $\bar{x}_i \leq \bar{y}_i(k) \leq \bar{x}_i \cdot (1 + \delta_k)$ (the left inequality stems from the fact that $\bar{y}(k)$ is an enclosure). Since we know that for our systems of linear equations, the actual upper bounds $\bar{x}_1, \ldots, \bar{x}_p$ are equal, we can take the smallest of the computed bounds

$$\bar{y}_{\min}(k) = \min_{1 \leq i \leq p} \bar{y}_i(k)$$

as the enclosure for all these upper bounds $\bar{x}_i$, $1 \leq i \leq p$. From

$$\bar{x}_i \leq \bar{y}_{\min}(k) \leq \bar{x}_i \cdot (1 + \delta_k) \tag{8}$$

we can conclude that

$$\frac{\bar{y}_{\min}(k)}{1 + \delta_k} \leq \bar{x}_i \leq \bar{y}_{\min}(k).$$

Let us choose $\varepsilon \in (0, 1)$ (e.g., $\varepsilon = 1/2$), and apply the algorithm $\mathcal{U}(k)$ to the auxiliary system $A^I\tilde{x} = \tilde{b}^I$ with

$$\mu = \mu_k = \bar{y}_{\min}(k) \cdot \frac{1 - \varepsilon}{1 + \delta_k}.$$

As a result, we get an enclosure $\overline{\overline{y}}$ for this auxiliary system, i.e., a vector for which

$$\overline{\overline{x}}_i \le \overline{\overline{y}}_i. \tag{9}$$

The actual upper bound $\overline{\overline{x}}_i$ for this new system is related to the actual upper bound $\overline{x}_i$ for the original system by a formula (7). Let us first show that the actual upper bound $\overline{\overline{x}}_i$ is non-zero. Indeed, from (8) and from the definition of μ_k, we conclude that

$$\mu_k = \overline{y}_{\min}(k) \cdot \frac{1-\varepsilon}{1+\delta_k} \le \overline{x}_i(1+\delta_k) \cdot \frac{1-\varepsilon}{1+\delta_k} = \overline{x}_i(1-\varepsilon) < \overline{x}_i$$

(for the last inequality, we used the fact that $\varepsilon < 1$). Therefore, $\overline{\overline{x}}_i = \overline{x}_i - \mu_k > 0$ for $i \le p$. Hence, for this system, the result $\overline{\overline{y}}$ of applying the algorithm $\mathcal{U}(\|)$ satisfies the inequality

$$|\overline{\overline{y}}_i - \overline{\overline{x}}_i| \le \delta_k \cdot \overline{\overline{x}}_i. \tag{10}$$

Adding μ_k to both sides of (9), we conclude that $\overline{x}_i \le \overline{y}_i(k+1)$, where we denoted $\overline{y}_i(k+1) = \overline{\overline{y}}_i + \mu_k$. Therefore, $\overline{y}_i(k+1)$ is indeed an enclosure (for $i \le p$). Let us find the value δ_{k+1} that corresponds to this new enclosure.

Since $\overline{x}_i = \overline{\overline{x}}_i + \mu_k$ and $\overline{y}_i(k+1) = \overline{\overline{y}}_i + \mu_k$, we conclude that $\overline{\overline{y}}_i - \overline{\overline{x}}_i = \overline{y}_i(k+1) - \overline{x}_i$, and therefore, (10) lead to the inequality

$$|\overline{y}_i(k+1) - \overline{x}_i| \le \delta_k \cdot \overline{\overline{x}}_i. \tag{11}$$

So, to estimate δ_{k+1}, we must estimate $\overline{\overline{x}}_i = \overline{x}_i - \mu_k$. From $\overline{y}_{\min}(k) \ge \overline{x}_i$ and the definition of μ_k, we have

$$\mu_k \ge \overline{x}_i \cdot \frac{1-\varepsilon}{1+\delta_k}.$$

Therefore,

$$\overline{\overline{x}}_i = \overline{x}_i - \mu_k \le \overline{x}_i \cdot \left(1 - \frac{1-\varepsilon}{1+\delta_k}\right).$$

Hence, from (11), we get the desired inequality

$$\left|\frac{\overline{y}_i(k+1) - \overline{x}_i}{\overline{x}_i}\right| \le \delta_{k+1}$$

with

$$\delta_{k+1} = \delta_k \cdot \left(1 - \frac{1-\varepsilon}{1+\delta_k}\right).$$

Estimating the number of computation steps. Each algorithm $\mathcal{U}(k+1)$ consists of two applications of an algorithm $\mathcal{U}(k)$. Therefore, the algorithm $\mathcal{U}(k)$ consists of 2^k applications of $\mathcal{U}$. So, to estimate the running time of this algorithm, we must estimate k.

Since $0 < 1 - \varepsilon < 1 + \delta_k$, we conclude that

$$0 < 1 - \frac{1-\varepsilon}{1+\delta_k} < 1$$

and therefore, that $\delta_{k+1} < \delta_k$. So, the sequence δ_k is decreasing. Hence, $\delta_k \le \delta_1 = \delta$ for all k. Therefore, $1 + \delta_k \le 1 + \delta$, and

$$1 - \frac{1-\varepsilon}{1+\delta_k} \le 1 - \frac{1-\varepsilon}{1+\delta}.$$

Hence, for every k,

$$\delta_{k+1} \le \delta_k \cdot \left(1 - \frac{1-\varepsilon}{1+\delta}\right).$$

So,

$$\delta_k \le \delta \cdot \left(1 - \frac{1-\varepsilon}{1+\delta}\right)^{k-1}.$$

To get $\delta_k \le 4/n^2$, we thus need k for which

$$\delta \cdot \left(1 - \frac{1-\varepsilon}{1+\delta}\right)^{k-1} \le \frac{4}{n^2}.$$

Applying binary logarithm to both sides of this inequality, we get

$$\log(\delta) + (k-1) \cdot \log\left(1 - \frac{1-\varepsilon}{1+\delta}\right) \le 2 - 2 \cdot \log(n)$$

so, it is sufficient to take $k \sim c_1 \log(n) + c_2$ for some constants c_i. For this k, the total number of applications of $\mathcal{U}$ is $2^k \sim 2^{c_1 \log(n) + c_2} = 2^{c_2} \cdot n^{c_1}$, i.e., it is polynomial in n. Therefore, the total algorithm that computes the $4/n^2$-accurate enclosure, is time-polynomial. Hence, due to lemma, P = NP. □

Proof of Theorems 2 and 2'. Suppose that for some $\delta(n) > 0$, there exists a polynomial-time algorithm $\mathcal{U}$ that for each strongly regular $n \times n$ interval matrix A^I and for each interval vector b^I computes a rational enclosure $[\underline{y}, \overline{y}]$ of the solution X satisfying $|\overline{y}_i - \overline{x}_i| \le \delta(n)$ for all i.

Let M be an arbitrary $p \times p$ MC-matrix, and let $A^I x = b^I$ be an $n \times n$ ($n = 3p$) interval system constructed in the lemma. Then, for every positive real number $\lambda > 0$, we can consider a new interval system $A^I \tilde{x} = \tilde{b}^I$, where $\tilde{b}^I = \lambda b^I$. We did not change the matrix A^I, so this matrix is still strongly regular. A vector x belongs to the solution set X of the original system iff the vector $\tilde{x} = \lambda x$ belongs to the solution set $\tilde{X}$ of the new system. Therefore, the optimal enclosure $\bar{\tilde{x}}$ for the new system is related to the optimal enclosure $\overline{x}$ for the original system $A^I x = b^I$ by a simple formula: $\bar{\tilde{x}} = \lambda \cdot \overline{x}$. Let us apply the algorithm $\mathcal{U}$ to the new system. As a result, we get an enclosure $\bar{\tilde{y}}$ for which

$$|\bar{\tilde{y}}_i - \bar{\tilde{x}}_i| = |\bar{\tilde{y}}_i - \lambda \cdot \overline{x}_i| \le \delta(n).$$

Dividing both sides of this in equality by λ, and denoting $\overline{y} = (1/\lambda)\bar{\tilde{y}}$, we conclude that

$$|\overline{y}_i - \overline{x}_i| \le \frac{\delta(n)}{\lambda}. \tag{12}$$

For our original interval system, for $i \le p$, we have $\overline{x}_i \ge 1/(2p^2)$ (this inequality, proven in [10], is part of what we have called our Main Lemma). Since $p = n/3$, we have

$$\overline{x}_i \ge \frac{1}{2(n/3)^2} = \frac{9}{2n^2}.$$

Therefore, for these i, we have $1 \le (2/9)n^2\overline{x}_i$. Multiplying both sides of this inequality by $\delta(n)/\lambda$, we can conclude that

$$\frac{\delta(n)}{\lambda} \le \frac{\delta(n)}{\lambda} \cdot \frac{2}{9} n^2 \overline{x}_i$$

and therefore, from (12), we can conclude that

$$|\overline{y}_i - \overline{x}_i| \le \frac{\delta(n)}{\lambda} \cdot \frac{2}{9} n^2 \overline{x}_i.$$

For $\lambda = \delta(n) \cdot n^4/18$, the right hand side side of this inequality turns into $(4/n^2)\overline{x}_i$. So, if we first apply the polynomial-time algorithm $\mathcal{U}$ to a system $A^I \tilde{x} = \tilde{b}^I$ with this λ, and then compute divide the resulting enclosures $\overline{\tilde{y}}_i$ by λ, we get the enclosures $\overline{y}_i$ for the original system $A^I x = b^I$ that satisfy the inequality

$$\left| \frac{\overline{y}_i - \overline{x}_i}{\overline{x}_i} \right| \le \frac{4}{n^2}$$

for $i \le p$. We have computed this new enclosure $\overline{y}_i$ in polynomial time, so, from lemma, it follows that P = NP. □

3. The symmetric case

Let $A^I = [A_c - \Delta, A_c + \Delta]$ be a symmetric interval matrix (i.e., the bounds $A_c - \Delta$ and $A_c + \Delta$ are symmetric) and let X^s be the set of solutions of (1) corresponding to systems with symmetric matrices only:

$$X^s = \{x;\ Ax = b \text{ for some } A \in A^I,\ b \in b^I,\ A \text{ symmetric}\}.$$

Enclosure methods for the symmetric case were given by Jansson [4] and Alefeld and Mayer [1].

J. Rohn has shown [10]:

- that computing the exact bounds

$$\underline{x}_i^s = \min_{X^s} x_i, \qquad \text{and} \qquad \overline{x}_i^s = \max_{X^s} x_i$$

is an NP-hard problem, and

- that computing $(4/n^2)$-accurate enclosures is also NP-hard.

In this paper, we will show that for every $\delta > 0$, computing δ-accurate enclosures is NP-hard. Formally, $[\underline{y}, \overline{y}]$ is called an enclosure of X^s if $X^s \subseteq [\underline{y}, \overline{y}]$ holds.

Theorem 3. *Suppose that for some $\delta > 0$, there exists a polynomial-time algorithm which for each strongly regular symmetric $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X^s satisfying*

$$\left| \frac{\overline{y}_i - \overline{x}_i^s}{\overline{x}_i^s} \right| \le \delta$$

for each i with $\overline{x}_i^s \ne 0$. Then P = NP.

Theorem 4. *Suppose that for some $\delta > 0$, there exists a polynomial-time algorithm which for each strongly regular symmetric $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X^s satisfying the inequality $|\overline{y}_i - \overline{x}_i^s| \le \delta$ for all i. Then* P = NP.

Theorem 4′. *Suppose that for some polynomial $\delta(n) > 0$, there exists a polynomial-time algorithm which for each strongly regular symmetric $n \times n$ interval matrix A^I and each b^I (both with rational bounds) computes a rational enclosure $[\underline{y}, \overline{y}]$ of X^s satisfying the inequality $|\overline{y}_i - \overline{x}_i^s| \leq \delta$ for all i. Then* P = NP.

Proof. The system (4)–(5) constructed in the Main Lemma has a symmetric interval matrix A^I and each $\overline{x}_i$, $i = 1, \ldots, n$, is achieved at the solution of a system whose matrix is of the form

$$\begin{pmatrix} 0 & -I & 0 \\ -I & 0 & M^{-1} \\ 0 & M^{-1} & M^{-1} - \beta z z^T \end{pmatrix}$$

(this is part of what we have called the Main Lemma), hence it is symmetric (since an MC-matrix M is symmetric). Thus we have

$$\overline{x}_i = \overline{x}_i^s \tag{13}$$

for $i = 1, \ldots, n$, and the proofs of Theorems 1, 2, and 2′ apply to this case as well. □

References

[1] Alefeld, G. and Mayer, G. *On the symmetric and unsymmetric solution set of interval systems.* SIAM J. Matr. Anal. Appl. **16** (1995), pp. 1223–1240.

[2] Alefeld, G. and Herzberger, J. *Introduction to interval computations.* Academic Press, N.Y., 1983.

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

[4] Jansson, C. *Interval linear systems with symmetric matrices, skew-symmetric matrices and dependencies in the right hand side.* Computing **46** (1991), pp. 265–274.

[5] Kreinovich, V., Lakeyev, A. V., and Noskov, S. I. *Optimal solution of interval linear systems is intractable (NP-hard).* Interval Computations 1 (1993), pp. 6–14.

[6] Kreinovich, V., Lakeyev, A. V., and Noskov, S. I. *Approximate linear algebra is intractable.* Linear Algebra and its Applications **232** (1) (1996), pp. 45–54.

[7] Lakeyev, A. V. and Kreinovich, V. *If input intervals are small enough, then interval computations are almost always easy.* In: "Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995". Reliable Computing (1995), Supplement, pp. 134–139.

[8] Neumaier, A. *Interval methods for systems of equations.* Cambridge University Press, Cambridge, 1990.

[9] Rohn, J. *Linear interval equations: computing sufficiently accurate enclosures is NP-hard.* Techn. Rep. No. 621, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 1995.

[10] Rohn, J. *Linear interval equations: computing enclosures with bounded relative overestimation is NP-hard.* In: Kearfott, R. B. and Kreinovich, V. (eds) "Applications of Interval Computations", Kluwer, Boston, MA, 1996, pp. 81–89.

[11] Rohn, J. and Kreinovich, V. *Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard.* SIAM J. Matr. Anal. Appl. **16** (1995), pp. 415–420.

Received: July 12, 1995
Revised version: February 28, 1996

V. Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968
E-mail: vladik@cs.utep.edu

A. V. Lakeyev
Irkutsk Computing Center
Russian Academy of Science
Siberian Branch
Lermontov Str. 134
Irkutsk 664033, Russia
Email: lakeyev@icc.ru