An Algorithm for Computing the Hull of the Solution Set of Interval Linear Equations

Rohn, Jiří
2010

Institute of Computer Science
Academy of Sciences of the Czech Republic

An Algorithm for Computing the Hull of the Solution Set of Interval Linear Equations

Jiří Rohn

Technical report No. V-1074

22.04.2010

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 051 111, fax: +420 286 585 789, e-mail:rohn@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

An Algorithm for Computing the Hull of the Solution Set of Interval Linear Equations

Jiří Rohn[1]

Technical report No. V-1074

22.04.2010

Abstract:

Described is a not-a-priori-exponential algorithm which for each $n \times n$ interval matrix $\mathbf{A}$ and for each interval n-vector $\mathbf{b}$ in a finite number of steps either computes the interval hull of the solution set of the system of interval linear equations $\mathbf{A}x = \mathbf{b}$, or finds a singular matrix $S \in \mathbf{A}$.

Keywords:
Interval linear equations, solution set, interval hull, algorithm, absolute value inequality.

[1]Supported by the Czech Republic Grant Agency under grant 201/09/1957 and by the Institutional Research Plan AV0Z10300504.

1 Introduction

This paper is dedicated to one of the most classical problems in interval analysis, namely computation of the interval hull of the solution set of a system of interval linear equations (which is an NP-hard problem). After introducing the notations being used in Section 2, we describe in Section 3 the problem itself and some results relevant to it. In Section 4 we quote a basic underlying result for computing enclosures of the solution set, which is then improved in Section 5 to yield instead the interval hull itself. In the following Section 6 we explain how to solve efficiently the absolute value matrix equations formulated in the main result of Section 5, and in Section 7 we give a detailed MATLAB-style description of the algorithm which, as the reader will see, is rather complex, but according to this author's experience, also efficient, and has been included into the INTERVALHULL.P file of the free software package VERSOFT [8].

2 Notations

We use the following notations. The ith row of a matrix A is denoted by $A_{i\bullet}$, the jth column by $A_{\bullet j}$. Matrix inequalities, as $A \le B$ or $A < B$, are understood componentwise. The absolute value of a matrix $A = (a_{ij})$ is defined by $|A| = (|a_{ij}|)$. The same notations also apply to vectors that are considered one-column matrices. Minimum or maximum of a finite set of vectors is also understood componentwise. I is the unit matrix, e_i is the ith column of I, and $e = (1, \dots, 1)^T$ is the vector of all ones. $Y_n = \{y \mid |y| = e\}$ is the set of all ± 1-vectors in $\mathbb{R}^n$, so that its cardinality is 2^n. Vectors $y, z \in Y_n$ are called adjacent if they differ in exactly one entry. Obviously, $y, z \in Y_n$ are adjacent if and only if $y = z - 2z_j e_j$ for some j. For each $x \in \mathbb{R}^n$ we define its sign vector $\operatorname{sgn}(x)$ by

$$(\operatorname{sgn}(x))_i = \begin{cases} 1 & \text{if } x_i \ge 0, \\ -1 & \text{if } x_i < 0 \end{cases} \qquad (i = 1, \dots, n),$$

so that $\operatorname{sgn}(x) \in Y_n$. For each $z \in \mathbb{R}^n$ we denote

$$T_z = \operatorname{diag}(z_1, \dots, z_n) = \begin{pmatrix} z_1 & 0 & \dots & 0 \\ 0 & z_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z_n \end{pmatrix},$$

and $\mathbb{R}^n_z = \{x \mid T_z x \ge 0\}$ is the orthant prescribed by the ± 1-vector $z \in Y_n$. An interval matrix is a set of matrices

$$\mathbf{A} = \{\, A \mid |A - A_c| \le \Delta \,\} = [A_c - \Delta, A_c + \Delta],$$

and an interval vector is a one-column interval matrix

$$\mathbf{b} = \{\, b \mid |b - b_c| \le \delta \,\} = [b_c - \delta, b_c + \delta].$$

3 The problem

Given an $n \times n$ interval matrix $\mathbf{A} = [A_c - \Delta, A_c + \Delta]$ and an interval n-vector $\mathbf{b} = [b_c - \delta, b_c + \delta]$, the solution set of the system of interval linear equations $\mathbf{A}x = \mathbf{b}$ is defined as

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) = \{ x \mid Ax = b \text{ for some } A \in \mathbf{A},\ b \in \mathbf{b} \}.$$

The Oettli-Prager theorem [5] asserts that the solution set is described by

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) = \{ x \mid |A_c x - b_c| \le \Delta |x| + \delta \}.$$

If $\mathbf{A}$ is regular (i.e., each $A \in \mathbf{A}$ is nonsingular), then $\mathbf{X}(\mathbf{A}, \mathbf{b})$ is compact and connected (Beeck [1]); if $\mathbf{A}$ is singular (i.e., it contains a singular matrix), then each component of $\mathbf{X}(\mathbf{A}, \mathbf{b})$ is unbounded (Jansson [3]). The solution set is generally of a complicated nonconvex structure. In practical computations, therefore, we look for an *enclosure* of it, i.e., for an interval vector $\mathbf{x}$ satisfying

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}.$$

If $\mathbf{A}$ is regular, then the intersection of all enclosures of $\mathbf{X}(\mathbf{A}, \mathbf{b})$ forms an interval vector which is called the *interval hull* (sometimes simply "hull") of $\mathbf{X}(\mathbf{A}, \mathbf{b})$ and is denoted by $\mathbf{x}(\mathbf{A}, \mathbf{b})$. Obviously,

$$\mathbf{x}(\mathbf{A}, \mathbf{b}) = [\underline{x}, \overline{x}],$$

where

$$\underline{x}_i = \min\{ x_i \mid x \in \mathbf{X}(\mathbf{A}, \mathbf{b}) \}, \tag{3.1}$$

$$\overline{x}_i = \max\{ x_i \mid x \in \mathbf{X}(\mathbf{A}, \mathbf{b}) \} \tag{3.2}$$

($i = 1, \ldots, n$). If $\mathbf{A}$ is singular, then $\mathbf{X}(\mathbf{A}, \mathbf{b})$ is either empty (a rare case), or unbounded and the interval hull is not defined in this case.

Computing the interval hull is NP-hard (original proof by Rohn and Kreinovich in [11], simplified proof in [2], Theorem 2.38). Exponential algorithms for its computation were suggested by Nickel [4] and Rohn [6]. Jansson [3] was first to propose the idea of "going along the orthants" having a nonempty intersection with $\mathbf{X}(\mathbf{A}, \mathbf{b})$, and bound the intersections, which are convex polytopes, by a linear programming technique. His method has the additional advantage that it does not require verification of regulariy/singularity in advance because it is verified (or disproved) on the way. In this paper we describe an algorithm based on a modification of his going-along-the-orthants idea which, however, employs a new technique of enclosing the intersections of the solution set with orthants by means of solutions of certain absolute value matrix equations. The main advantage of this approach consists in the fact that it allows for a verified implementation (where the result is true despite being achieved by means of finite precision arithmetic). This implementation was done in the INTERVALHULL.P file of the freely available verification software VERSOFT [8] written in INTLAB, a MATLAB toolbox developed by Rump [12]. INTERVALHULL, which was created between 2000 and 2008 and whose source file has about 1000 lines, is available there as a p-coded file only; here we reveal the algorithm behind it for the first time.

4 Underlying result

The starting point of our considerations is the following result which has been proved in [10, Thm. 3] (without using $\overline{x}_z$ and $\underline{x}_z$ that are introduced here in (4.3), (4.4), so that the original formulation was more cumbersome).

Theorem 1. *Let* $\mathbf{A} = [A_c - \Delta, A_c + \Delta]$ *be an* $n \times n$ *interval matrix,* $\mathbf{b} = [b_c - \delta, b_c + \delta]$ *an interval* n*-vector, and let* Z *be a subset of* Y_n *having the following properties:*

(a) $\operatorname{sgn}(x_0) \in Z$ *for some* $x_0 \in \mathbf{X}(\mathbf{A}, \mathbf{b})$,

(b) for each $z \in Z$ *the inequalities*

$$\begin{aligned} (QA_c - I)T_z &\geq |Q|\Delta, & (4.1)\\ (QA_c - I)T_{-z} &\geq |Q|\Delta & (4.2) \end{aligned}$$

have matrix solutions Q_z *and* Q_{-z}*, respectively; denote*

$$\begin{aligned} \overline{x}_z &= Q_z b_c + |Q_z|\delta, & (4.3)\\ \underline{x}_z &= Q_{-z} b_c - |Q_{-z}|\delta, & (4.4) \end{aligned}$$

(c) if $z \in Z$, $\underline{x}_z \leq \overline{x}_z$, *and* $(\underline{x}_z)_j(\overline{x}_z)_j \leq 0$ *for some* j*, then* $z - 2z_j e_j \in Z$.

Then $\mathbf{A}$ *is regular and*

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) \subseteq [\min_{z \in Z_1} \underline{x}_z, \max_{z \in Z_1} \overline{x}_z] \tag{4.5}$$

holds, where

$$Z_1 = \{\, z \in Z \mid \underline{x}_z \leq \overline{x}_z \,\}.$$

There are several facts, established in [10], needed for understanding the procedure involved:

(i) the set Z of ± 1-vectors z representing the orthants $\mathbb{R}^n_z$ is constructed inductively in steps (a) and (c),

(ii) $\mathbf{X}(\mathbf{A}, \mathbf{b}) \cap \mathbb{R}^n_z = \emptyset$ for each $z \in Z - Z_1$,

(iii) $\mathbf{X}(\mathbf{A}, \mathbf{b}) \cap \mathbb{R}^n_z \subseteq [\underline{x}_z, \overline{x}_z]$ for each $z \in Z_1$,

(iv) if the procedure of constructing Z is brought to conclusion (i.e., all the Q_z's and Q_{-z}'s needed have been found), then

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) \subseteq \bigcup_{z \in Z_1} [\underline{x}_z, \overline{x}_z]. \tag{4.6}$$

In the main result of this paper we show that if the inequalities (4.1), (4.2) are always solved as equations, then the enclosure in (4.5) becomes the interval hull of $\mathbf{X}(\mathbf{A}, \mathbf{b})$.

5 Main result

Theorem 2. *Let* $\mathbf{A} = [A_c - \Delta, A_c + \Delta]$ *be an* $n \times n$ *interval matrix,* $\mathbf{b} = [b_c - \delta, b_c + \delta]$ *an interval* n*-vector, and let* Z *be a subset of* Y_n *having the following properties:*

(a) $\operatorname{sgn}(x_0) \in Z$ *for some* $x_0 \in \mathbf{X}(\mathbf{A}, \mathbf{b})$*,*

(b') for each $z \in Z$ *the inequalities*

$$\begin{aligned} QA_c - |Q|\Delta T_z &= I, \qquad (5.1)\\ QA_c - |Q|\Delta T_{-z} &= I \qquad (5.2)\end{aligned}$$

have matrix solutions Q_z *and* Q_{-z}*, respectively; denote*

$$\begin{aligned} \overline{x}_z &= Q_z b_c + |Q_z|\delta, \qquad (5.3)\\ \underline{x}_z &= Q_{-z} b_c - |Q_{-z}|\delta, \qquad (5.4)\end{aligned}$$

(c) if $z \in Z$*,* $\underline{x}_z \le \overline{x}_z$*, and* $(\underline{x}_z)_j (\overline{x}_z)_j \le 0$ *for some* j*, then* $z - 2z_j e_j \in Z$*.*

Then $\mathbf{A}$ *is regular and*

$$\mathbf{x}(\mathbf{A}, \mathbf{b}) = [\min_{z \in Z_1} \underline{x}_z, \max_{z \in Z_1} \overline{x}_z] \qquad (5.5)$$

holds, where

$$Z_1 = \{ z \in Z \mid \underline{x}_z \le \overline{x}_z \}.$$

Proof. If Q_z and Q_{-z} solve (5.1) and (5.2), respectively, then they satisfy

$$\begin{aligned} (Q_z A_c - I)T_z &= |Q_z|\Delta, \\ (Q_{-z} A_c - I)T_{-z} &= |Q_{-z}|\Delta, \end{aligned}$$

so that they are solutions of (4.1), (4.2). Thus, the assumption (b') implies validity of the assumption (b) of Theorem 1, hence all three assumptions of Theorem 1 are met and consequently $\mathbf{A}$ is regular and

$$\mathbf{X}(\mathbf{A}, \mathbf{b}) \subseteq [\min_{z \in Z_1} \underline{x}_z, \max_{z \in Z_1} \overline{x}_z] \qquad (5.6)$$

holds. Denote

$$\begin{aligned} [\underline{x}, \overline{x}] &= \mathbf{x}(\mathbf{A}, \mathbf{b}), \qquad (5.7)\\ [\underline{\underline{x}}, \overline{\overline{x}}] &= [\min_{z \in Z_1} \underline{x}_z, \max_{z \in Z_1} \overline{x}_z]. \qquad (5.8)\end{aligned}$$

In view of (5.6), $[\underline{\underline{x}}, \overline{\overline{x}}]$ is an enclosure of $\mathbf{X}(\mathbf{A}, \mathbf{b})$ and since $\mathbf{x}(\mathbf{A}, \mathbf{b})$ is the intersection of all the enclosures, we have

$$[\underline{x}, \overline{x}] \subseteq [\underline{\underline{x}}, \overline{\overline{x}}],$$

in particular

$$\overline{x} \leq \overline{\overline{x}}. \tag{5.9}$$

Take an $i \in \{1, \dots, n\}$. Then (5.9) implies $\overline{x}_i \leq \overline{\overline{x}}_i$. To prove the converse inequality, let $z \in Z_1$ be such that

$$\overline{\overline{x}}_i = (\overline{x}_z)_i$$

(by (5.8)), and define

$$q^T = e_i^T Q_z$$

and

$$y = \operatorname{sgn}(q),$$

so that $|q| = T_y q$. Premultiplying the equation (5.1) by e_i^T, we get

$$q^T A_c - |q|^T \Delta T_z = q^T A_c - q^T T_y \Delta T_z = q^T (A_c - T_y \Delta T_z) = e_i^T,$$

hence

$$q^T = e_i^T (A_c - T_y \Delta T_z)^{-1}$$

(since y, z are ± 1-vectors, so that $A_c - T_y \Delta T_z \in \mathbf{A}$, and we have already proved that $\mathbf{A}$ is regular). Now we have

$$\overline{\overline{x}}_i = (\overline{x}_z)_i = e_i^T (Q_z b_c + |Q_z|\delta) = q^T b_c + q^T T_y \delta = e_i^T (A_c - T_y \Delta T_z)^{-1} (b_c + T_y \delta) = x_i,$$

where x is the solution of

$$(A_c - T_y \Delta T_z) x = b_c + T_y \delta$$

with $A_c - T_y \Delta T_z \in \mathbf{A}$ (as we already know) and $b_c + T_y \delta \in \mathbf{b}$ (since y is a ± 1-vector). This shows that $x \in \mathbf{X}(\mathbf{A}, \mathbf{b})$, hence $\overline{\overline{x}}_i$ is attained over $\mathbf{X}(\mathbf{A}, \mathbf{b})$, and in view of (3.2) this means that $\overline{\overline{x}}_i \leq \overline{x}_i$, so that $\overline{\overline{x}}_i = \overline{x}_i$. Since i was arbitrary, we finally obtain that

$$\overline{\overline{x}} = \overline{x}.$$

The equality $\underline{\underline{x}} = \underline{x}$ is proved in a similar way. This shows that

$$\mathbf{x}(\mathbf{A}, \mathbf{b}) = [\underline{\underline{x}}, \overline{\overline{x}}],$$

which completes the proof. $\square$

6 Computation of the matrices Q_z

In the light of the main result, what remains to be resolved is the problem of solving the matrix equation

$$QA_c - |Q|\Delta T_z = I \tag{6.1}$$

(and the related equation (5.2) which is of the same form). Take an $i \in \{1, \dots, n\}$ and put

$$x^T = Q_{i\bullet},$$

then x solves

$$x^T A_c - |x|^T \Delta T_z = e_i^T$$

and thus also

$$A_c^T x - T_z \Delta^T |x| = e_i. \tag{6.2}$$

The equation (6.2) is an equation of the form

$$Ax + B|x| = b \tag{6.3}$$

($A, B \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$), which is called an *absolute value equation*. In [7], [9] we have developed a very efficient algorithm **absvaleqn** for its solution, reproduced here in Fig. 7.2 in a MATLAB-like form. It is called by

$$[x, S] = \textbf{absvaleqn}(A, B, b)$$

and its main feature is that for each data A, B, b, it produces in a finite number of steps either a solution x of the equation (6.3), or a singular matrix S satisfying $|S - A| \leq |B|$ (but not both; the unassigned variable is outputted as [ ], an empty matrix/vector).Thus, our equation (6.2) can be solved by

$$[x, S] = \textbf{absvaleqn}(A_c^T, -T_z\Delta^T, e_i)$$

and the output is either a vector x with $x^T = (Q_z)_{i\bullet}$, or a singular matrix S satisfying

$$|S - A_c^T| \leq |-T_z\Delta^T| = \Delta^T,$$

so that

$$|S^T - A_c| \leq \Delta$$

and S^T is a singular matrix in $\mathbf{A}$. In this way we may compute the matrix Q_z row-by-row. This procedure is formalized in the MATLAB-like function **qzmatrix** placed in Fig. 7.1 as a subfunction of the main function **intervalhull** to be explained in the next section. From what has been said above it follows that *the function* **qzmatrix** *called by*

$$[Q_z, S] = \textbf{qzmatrix}(\mathbf{A}, z)$$

in a finite number of steps either finds a solution Q_z of the matrix equation (6.1), or produces a singular matrix $S \in \mathbf{A}$.

7 The algorithm

The algorithm **intervalhull**, described in MATLAB-like style in Figs. 7.1 and 7.2, consists of the main function **intervalhull** and of two subfunctions **qzmatrix** and **absvaleqn** (the headers of the (sub)functions are marked in blue, and their calls in green). The main function **intervalhull** closely copies Theorem 2, the subfunction **qzmatrix** is based on the results of Section 6, and the subfunction **absvaleqn** comes from [9]. Since the subfunction **absvaleqn** terminates in a finite number of steps (Theorem 3 in [9]) and the same thus holds for **qzmatrix**, we have this finite termination theorem.

Theorem 3. *For each $n \times n$ interval matrix $\mathbf{A}$ and for each interval n-vector $\mathbf{b}$ the algorithm* **intervalhull** *(Figs. 7.1, 7.2) in a finite number of steps either computes the interval hull $\mathbf{x}$ of the solution set of the system of interval linear equations $\mathbf{A}x = \mathbf{b}$, or produces a singular matrix $S \in \mathbf{A}$ (but not both).*

The algorithm is not-a-priori-exponential since it requires $2n \cdot \mathrm{card}(Z)$ calls of the subfunction **absvaleqn** which itself takes about $0.1 \cdot n$ steps on the average [7]. For example, if the solution set $\mathbf{X}(\mathbf{A}, \mathbf{b})$ is a part of the interior of a single orthant, then only $2n$ calls of **absvaleqn** are made.

8 Implementation

The algorithm has been implemented in the VERINTERVALHULL.M function of the freely available verification software package VERSOFT [8].

function $[\mathbf{x}, S] =$ **intervalhull** $(\mathbf{A}, \mathbf{b})$
% Computes either the interval hull $\mathbf{x}$
% of the solution set of $\mathbf{A}x = \mathbf{b}$,
% or a singular matrix $S \in \mathbf{A}$.
$\mathbf{x} = [\,]$; $S = [\,]$;
if A_c is singular, $S = A_c$; **return**, **end**
$x_c = A_c^{-1} b_c$; $z = \mathrm{sgn}(x_c)$; $\underline{x} = x_c$; $\overline{x} = x_c$;
$Z = \{z\}$; $D = \emptyset$;
while $Z \neq \emptyset$
  select $z \in Z$; $Z = Z - \{z\}$; $D = D \cup \{z\}$;
  $[Q_z, S] =$ **qzmatrix** $(\mathbf{A}, z)$;
  if $S \neq [\,]$, $\mathbf{x} = [\,]$; **return**, **end**
  $[Q_{-z}, S] =$ **qzmatrix** $(\mathbf{A}, -z)$;
  if $S \neq [\,]$, $\mathbf{x} = [\,]$; **return**, **end**
  $\overline{x}_z = Q_z b_c + |Q_z|\delta$;
  $\underline{x}_z = Q_{-z} b_c - |Q_{-z}|\delta$;
  if $\underline{x}_z \leq \overline{x}_z$
    $\underline{x} = \min(\underline{x}, \underline{x}_z)$; $\overline{x} = \max(\overline{x}, \overline{x}_z)$;
    for $j = 1 : n$
      $z' = z$; $z'_j = -z'_j$;
      if $((\underline{x}_z)_j (\overline{x}_z)_j \leq 0$ **and** $z' \notin Z \cup D)$
        $Z = Z \cup \{z'\}$;
      end
    end
  end
end
$\mathbf{x} = [\underline{x}, \overline{x}]$;
%% *Subfunction*
function $[Q_z, S] =$ **qzmatrix** $(\mathbf{A}, z)$
% Computes either a solution Q_z
% of the equation $QA_c - |Q|\Delta T_z = I$,
% or a singular matrix $S \in \mathbf{A}$.
for $i = 1 : n$
  $[x, S] =$ **absvaleqn** $(A_c^T, -T_z \Delta^T, e_i)$;
  if $S \neq [\,]$, $S = S^T$; $Q_z = [\,]$; **return**
  end
  $(Q_z)_{i\bullet} = x^T$;
end
$S = [\,]$;
%% *Continuation: Subfunction absvaleqn*

Figure 7.1: An algorithm for computing the interval hull.

%% *Continuation: Subfunction absvaleqn*

function $[x, S] =$ **absvaleqn** (A, B, b)
% Finds either a solution x to $Ax + B|x| = b$, or
% a singular matrix S satisfying $|S - A| \le |B|$.
$x = [\,]$; $S = [\,]$; $i = 0$; $r = 0 \in \mathbb{R}^n$; $X = 0 \in \mathbb{R}^{n \times n}$;
if A is singular, $S = A$; **return**, **end**
$z = \mathrm{sgn}(A^{-1}b)$;
if $A + BT_z$ is singular, $S = A + BT_z$; **return**, **end**
$x = (A + BT_z)^{-1}b$;
$C = -(A + BT_z)^{-1}B$;
while $z_j x_j < 0$ for some j
    $i = i + 1$;
    $k = \min\{j \mid z_j x_j < 0\}$;
    if $1 + 2z_k C_{kk} \le 0$
        $S = A + B(T_z + (1/C_{kk})e_k e_k^T)$;
        $x = [\,]$; **return**
    end
    if $((k < n$ **and** $r_k > \max_{k<j} r_j)$ **or** $(k = n$ **and** $r_n > 0))$
        $x = x - X_{\bullet k}$;
        for $j = 1 : n$
            if $(|B||x|)_j > 0$, $y_j = (Ax)_j/(|B||x|)_j$; **else** $y_j = 1$; **end**
        end
        $z = \mathrm{sgn}(x)$;
        $S = A - T_y |B| T_z$;
        $x = [\,]$; **return**
    end
    $r_k = i$;
    $X_{\bullet k} = x$;
    $z_k = -z_k$;
    $\alpha = 2z_k/(1 - 2z_k C_{kk})$;
    $x = x + \alpha x_k C_{\bullet k}$;
    $C = C + \alpha C_{\bullet k} C_{k \bullet}$;
end

Figure 7.2: An algorithm for computing the interval hull, subfunction *absvaleqn*.

Bibliography

[1] H. Beeck, *Charakterisierung der Lösungsmenge von Intervallgleichungssystemen*, Zeitschrift für Angewandte Mathematik und Mechanik, 53 (1973), pp. T181–T182. 3

[2] M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann, *Linear Optimization Problems with Inexact Data*, Springer-Verlag, New York, 2006. 3

[3] C. Jansson, *Calculation of exact bounds for the solution set of linear interval systems*, Linear Algebra and Its Applications, 251 (1997), pp. 321–340. 3

[4] K. Nickel, *Die Ueberschätzung des Wertebereichs einer Funktion in der Intervallrechnung mit Anwendungen auf lineare Gleichungssysteme*, Computing, 18 (1977), pp. 15–36. 3

[5] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numerische Mathematik, 6 (1964), pp. 405–409. 3

[6] J. Rohn, *Systems of linear interval equations*, Linear Algebra and Its Applications, 126 (1989), pp. 39–78. 3

[7] J. Rohn, *An algorithm for solving the absolute value equation*, Electronic Journal of Linear Algebra, 18 (2009), pp. 589–599. http://www.math.technion.ac.il/iic/ela/ela-articles/articles/vol18_pp589-599.pdf. 7, 8

[8] J. Rohn, *VERSOFT: Verification software in MATLAB/INTLAB*, 2009. `http://www.cs.cas.cz/rohn/matlab`. 2, 3, 8

[9] J. Rohn, *An algorithm for solving the absolute value equation: An improvement*, Technical Report 1063, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, January 2010. http://uivtx.cs.cas.cz/~rohn/publist/absvaleqnreport.pdf. 7, 8

[10] J. Rohn, *A general method for enclosing solutions of interval linear equations*, Technical Report 1067, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, March 2010. http://uivtx.cs.cas.cz/~rohn/publist/genmeth.pdf. 4

[11] J. Rohn and V. Kreinovich, *Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard*, SIAM Journal on Matrix Analysis and Applications, 16 (1995), pp. 415–420. 3

[12] S. Rump, *INTLAB - INTerval LABoratory*, in Developments in Reliable Computing, T. Csendes, ed., Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104. http://www.ti3.tu-harburg.de/rump/. 3