Kalpa Publications in Computing

Volume 8, 2018, Pages 54–69

TNC'18. Trusted Numerical Computations

# Detecting Unsolvability of Interval Linear Systems

Jaroslav Horáček[1] [*], Jan Horáček[2], and Milan Hladík[3] [†]

[1] Department of Applied Mathematics, Charles University, Prague, Czech Republic.
horacek@kam.mff.cuni.cz
[2] Faculty of Informatics and Mathematics, University of Passau, Passau, Germany.
Jan.Horacek@uni-passau.de
[3] Department of Applied Mathematics, Charles University, Prague, Czech Republic.
hladik@kam.mff.cuni.cz

**Abstract**

In this paper we deal with detection of unsolvability of interval linear systems. Various methods based on existing algorithms or on existing sufficient conditions are developed. The methods are tested on a large variety of random systems and the results are visualized. The two strongest sufficient conditions are proved to be equivalent under a certain assumption. The topic of detecting solvability is also touched upon.

## 1 Introduction

There exist many methods for computing interval enclosures of the solution set of an interval linear system. Nevertheless, many of them return nonempty enclosure even if the system has no solution. In some applications such as system validation or technical computing, we do care whether the systems are solvable or unsolvable. In general, checking solvability and unsolvability are both hard problems [7, 22] (NP-complete and coNP-complete respectively). That is why it would be favorable to have at least some sufficient conditions or algorithms detecting unsolvability that are computable in polynomial time. In this work, such algorithms and conditions are discussed. Most of them are well-known, but used so far for a different purpose than checking unsolvability. We are going to show how they can be modified to detect unsolvability, what are the relations between them and how strong they are. First, let us start with some basic notation followed by basic definitions.

## 2 Basic Notation and Definitions

An interval matrix $\boldsymbol{A}$ of size $m \times n$ is defined as $\boldsymbol{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} \mid \underline{A} \leq A \leq \overline{A}\}$, where the inequality relation $\leq$ is understood component-wise. In the further text the relation $<$ is also understood component-wise. An interval vector is a special case of an interval matrix

[*] Jaroslav Horáček was supported by GAČR grant P403-18-04735S.
[†] Milan Hladík was supported by GAČR grant P403-18-04735S.

and is defined similarly. We denote interval structures in boldface (e.g., $\boldsymbol{A}, \boldsymbol{b}$). The point real structures are denoted using normal font (e.g., $A, b$). Two often used matrices corresponding to an interval matrix $\boldsymbol{A}$ are the *midpoint matrix*

$$A_c = \frac{\overline{A} + \underline{A}}{2},$$

and the *radius matrix*

$$A_\Delta = \frac{\overline{A} - \underline{A}}{2}.$$

An interval matrix $\boldsymbol{A}$ can be also defined using the midpoint and radius matrix as

$$\boldsymbol{A} = [A_c - A_\Delta, A_c + A_\Delta].$$

The coefficients of a real matrix $A$ and of an interval matrix $\boldsymbol{A}$ at the position $(i, j)$ are denoted by $A_{ij}$ and $\boldsymbol{A}_{ij}$ respectively. The absolute value of a real matrix is defined as $|A|_{ij} = |A_{ij}|$. The absolute value of an interval matrix is defined as

$$|\boldsymbol{A}|_{ij} = \max\{|\underline{A}_{ij}|, |\overline{A}_{ij}|\}.$$

The spectral radius of a real matrix $A$ is the largest absolute value of all eigenvalues and it is denoted by $\rho(A)$. We will also use the notation $I$ for an identity matrix of a corresponding size. The symbol $e$ denotes a vector $(1, \dots, 1)^T$ of a corresponding size. In the further text, the Moore-Penrose pseudoinverse of a matrix $A$ will be denoted by $A^+$. The following definitions are crucial for the next sections.

**Definition 1** (Solution set)**.** By the *solution set* of an interval linear system $\boldsymbol{A}x = \boldsymbol{b}$ we understand the set

$$\Sigma = \{x \mid Ax = b \text{ for some } A \in \boldsymbol{A}, b \in \boldsymbol{b}\}.$$

**Definition 2** (Unsolvability)**.** If the solution set $\Sigma$ of $\boldsymbol{A}x = \boldsymbol{b}$ is empty, we call the system *unsolvable*. In another words, no system $Ax = b$ in $\boldsymbol{A}x = \boldsymbol{b}$ has a solution.

We use the notion *square* interval linear system to address the case when the system matrix $\boldsymbol{A}$ is of size $n \times n$; by an *overdetermined* interval linear system we mean an interval linear system with more equations than variables (i.e., its matrix $\boldsymbol{A}$ is of size $m \times n$, where $m > n$). In this work we do not deal with underdetermined systems (the $m < n$ case).

The solution set $\Sigma$ is usually hard to be described (it is generally a set that is convex in each orthant). To simplify a calculation with the set $\Sigma$, its enclosure is often used.

**Definition 3** (Enclosure)**.** An $n$-dimensional box $\boldsymbol{x}$ (interval vector) is called an *enclosure* of the solution set $\Sigma$ if

$$\Sigma \subseteq \boldsymbol{x}.$$

If $\boldsymbol{x}$ is the tightest possible enclosure it is called the *hull*.

In general, computing the exact hull is an NP-hard problem [10], that is why in most of the applications only enclosures are used. Of course, the tighter the enclosures the better. When we refer to ”solving” a system we usually have computing of an enclosure in mind.

# 3 Conditions and Algorithms Detecting Unsolvability

Now, we are ready to explore individual conditions and algorithms. The mentioned methods are usually well-known, however, they are not often used for detecting unsolvability.

## 3.1 Linear Programming

We start with the famous theorem by Oettli-Prager describing the solution set of an interval linear system (see e.g., [18]).

**Theorem 1** (Oettli-Prager)**.** *A vector $x \in \mathbb{R}^n$ is a (weak) solution of an interval system $\boldsymbol{A}x = \boldsymbol{b}$ (i.e., $x$ is a solution of some $Ax = b$ within $\boldsymbol{A}x = \boldsymbol{b}$) if and only if*

$$|A_c x - b_c| \leq A_\Delta |x| + b_\Delta.$$

We can rewrite the first absolute value into two inequalities. For each orthant, the second absolute value can be easily dismissed because we know the sign of each component of $x$ lying in this orthant, hence we know its absolute value. Detailed information about rewriting the formulas can be seen in e.g., [5]. For each orthant, this adjustment results in a set of linear inequalities and therefore it can be solved by verified linear programming ( see e.g., [1, 9]). In an orthant we must solve two linear programming problems (for maximum and minimum) in each of $n$ directions. In total, $2n \times 2^n$ linear programing problems need to be solved. However, if there is a precomputed enclosure of the solution set, it may intersect only some orthants and hence reduce the total number of linear programming problems to be solved. Another favorable situation is when the number of variables $n$ is small.

If a verified linear programming announces nonexistence of a solution in all suspected orthants, the system is unsolvable. However, the verified linear programming might not always be able to decide about the existence of a solution in each orthant.

The computation time of this method might be too long. Moreover, the method requires an implementation of a verified linear programming. That is why we only mention this method for the sake of completeness, and we are not going to compare it against the other methods.

## 3.2 Interval Gaussian Elimination

Some famous algorithms for enclosing the solution set of an interval system can be also used to detect unsolvability. One of the methods is the interval version of Gaussian elimination. We are not going to mention the verified LU decomposition, since the method is mathematically similar to Gaussian elimination (for references to verified LU decomposition see e.g., [13, 19]). More detailed description of the interval Gaussian elimination can be found in [3] or [5]. Here we only present its main ideas. Let us have an interval system $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A}$ is an interval matrix of size $m \times n$. Simply said, we transform the system with row equivalent operations similarly as in the point real case. When we eliminate coefficients under the pivot, we set these coefficients directly to the intervals $[0, 0]$. These elimination steps can be interpreted as performing the classical elimination on all real systems contained in the interval system separately and subsequently gathering the resulting systems back into an interval system. Note that, there is no need to use overestimating interval arithmetics here. Nevertheless, for the elimination of the other coefficients we have to use interval arithmetics. We eliminate all columns of the system matrix except from the last one. The last $m - n + 1$ rows of the eliminated system are in the following shape:

$$\begin{aligned} \boldsymbol{c}_n x_n &= \boldsymbol{d}_n, \\ \boldsymbol{c}_{n+1} x_n &= \boldsymbol{d}_{n+1}, \\ &\vdots \\ \boldsymbol{c}_m x_n &= \boldsymbol{d}_m, \end{aligned}$$

for some intervals $\boldsymbol{c}_n, \ldots, \boldsymbol{c}_m, \boldsymbol{d}_n, \ldots, \boldsymbol{d}_m$, that occurred during the elimination. Now, the interval enclosure of the solution of the variable $x_n$ is

$$\boldsymbol{x}_n = (\boldsymbol{d}_n / \boldsymbol{c}_n) \cap \cdots \cap (\boldsymbol{d}_m / \boldsymbol{c}_m).$$

If such an intersection is empty, then the original system is unsolvable.

Gaussian elimination is often used with preconditioning. However, multiplying an interval system with a matrix usually results in overestimation of the original interval coefficients. When the preconditioning specified in [3] is used, an unsolvable system usually becomes solvable. However, if we do not use the preconditioning, the interval operations may result in an overestimation anyway, hence we get a solvable system again, even if the original one was unsolvable. That is why detection of unsolvability by interval Gaussian elimination is suitable only for very small systems. We are going to address the suitable sizes of systems for this method later.

## 3.3 Square Subsystems

Let us have an overdetermined interval system $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A}$ is of size $m \times n$ with $m > n$. The system that occurs by selecting some $n$ equations from the original system we call *square subsystem* (the matrix of the new system is now of size $n \times n$). The following lemma will be used to detect unsolvability.

**Lemma 1.** *Let $\Sigma$ be the solution set of $\boldsymbol{A}x = \boldsymbol{b}$ and let $\Sigma_{subsq}$ be the solution set of a square subsystem of $\boldsymbol{A}x = \boldsymbol{b}$. Then*

$$\Sigma \subseteq \Sigma_{subsq}.$$

*Proof.* The original system $\boldsymbol{A}x = \boldsymbol{b}$ has more equations that can put no or more restrictions on the solution set of the square subsystem. □

There exist many methods for computing enclosures of interval systems where the matrix $\boldsymbol{A}$ is of size $n \times n$ [4, 11, 14, 18, 25]. If more square subsystems are chosen and their enclosures computed, unsolvability could be easily detected. According to Lemma 1 the solution set of the original overdetermined system must lie inside the intersection of all these enclosures. If the intersection of these enclosures is empty, then the whole overdetermined system is not solvable. For a more sophisticated use of square subsystems we refer to our previous work [6].

## 3.4 Least Squares Enclosure

When an overdetermined system occurs, one often thinks of its least squares solution. Let us have an overdetermined point real system $Ax = b$. The goal of the least squares approach is to compute $x$ that minimizes the Euclidean norm $\|Ax - b\|$. It holds that $\|y\| = 0$ if and only if $y = 0$. Therefore, if $\|Ax - b\| = 0$, then also $Ax - b = 0$ and the system $Ax = b$ is solved by $x$. Now, let us move to the interval case. As usually, an interval system can be viewed as a set of point real systems. First, an enclosure of all least squares solutions $A^T Ax = A^T b$ for all $A \in \boldsymbol{A}, b \in \boldsymbol{b}$ is computed. Possibly, the best way to do that [17] is by solving the following system

$$\begin{pmatrix} I & \boldsymbol{A} \\ \boldsymbol{A}^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix}. \tag{1}$$

The enclosure of the all least squares solutions $\boldsymbol{x}$ appears as the first $n$ components of the obtained enclosure. If $0 \notin \boldsymbol{Ax} - \boldsymbol{b}$ we are sure that there is no $x, A, b$ such that $Ax - b = 0$, $A \in \boldsymbol{A}, b \in \boldsymbol{b}$ and the original interval system is not solvable. Another possibility to prove unsolvability, is to check whether $0 \notin \boldsymbol{y}$, where $\boldsymbol{y}$ appears as the last $m$ components of the obtained enclosure [2].

### 3.5 Full Column Rank

In this section we use full column rank sufficient conditions for detecting unsolvability of an interval linear system. We say that an $m \times n$ real matrix $A$ has *full column rank* if its rank is equal to the number of columns, i.e. $\operatorname{rank}(A) = n$. We say that an $m \times n$ interval matrix $\boldsymbol{A}$ has full column rank if every instance $A \in \boldsymbol{A}$ has full column rank.

Let $\boldsymbol{A}x = \boldsymbol{b}$ be an interval linear system, where $\boldsymbol{A}$ is of size $m \times n$. If for every instance $Ax = b$, where $A \in \boldsymbol{A}, b \in \boldsymbol{b}$ the matrix $(\, A \,|\, b \,)$ has full column rank, then it means that the vector $b$ does not belong to the column space of $A$ and hence the system has no solution (according to the well-known Frobenius theorem). Therefore, the whole interval system $\boldsymbol{A}x = \boldsymbol{b}$ is unsolvable, if every matrix $C \in (\, \boldsymbol{A} \,|\, \boldsymbol{b} \,)$ has full column rank.

Checking whether an interval matrix has full column rank is a coNP-complete problem [7]. However, the following sufficient condition mentioned in a handbook by Rohn [24] can be used. For a proof of this theorem we refer to [26].

**Theorem 2.** *Let us have an interval matrix $\boldsymbol{A} = [A_c - A_\Delta, A_c + A_\Delta]$ of size $m \times n$. If $A_c$ has full column rank and*

$$\rho(|A_c^+| \cdot A_\Delta) < 1, \tag{2}$$

*then $\boldsymbol{A}$ has full column rank.*

Since $A_c$ consists of linearly independent columns we can write

$$A_c^+ = \left(A_c^T A_c\right)^{-1} A_c^T.$$

In the further text, many of our results will be in terms of matrix norms. We will use only *submultiplicative* matrix norms i.e, those that satisfy

$$\|A \cdot B\| \leq \|A\| \cdot \|B\|$$

for real matrices (or vectors) $A, B$ of the corresponding size. In [23] the following theorem by Rohn can be found. We formulate it in a stronger way as an equivalence and we provide a simpler version of Rohn's proof.

**Theorem 3** (Rohn)**.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix. There exists a matrix $R \in \mathbb{R}^{n\times m}$ such that for an arbitrary submultiplicative matrix norm $\|\cdot\|$ the inequality*

$$\|I - RA\| < 1$$

*holds, if and only if $A$ has full column rank.*

*Proof.* ($\Leftarrow$) This implication is rather simple. If $A$ has full column rank then $A^TA$ is regular and therefore we can choose $R = (A^TA)^{-1}A^T$ and we know that $RA = I$ therefore $\|I - RA\| = 0 < 1$.

($\Rightarrow$) Let there be a matrix $R \in \mathbb{R}^{n\times m}$ such that

$$\|I - RA\| < 1.$$

Using the well-known relation between the spectral radius and a submultiplicative norm,

$$\rho(I - RA) \leq \|I - RA\| < 1.$$

Hence, $I - RA$ has all its eigenvalues located somewhere within a circle with the centre 0 and radius $< 1$. Adding $I$ to the matrix $-RA$ shifts all its eigenvalues to the right by 1. The matrix $-RA$ has all its eigenvalues located within a circle with the center $-1$ and radius $< 1$. This circle does not intersect 0, therefore, no eigenvalue can be 0 and therefore $-RA$ and also $RA$ are nonsingular. This implies that $A$ must have full column rank otherwise $RA$ would be singular. □

The remaining question is how to choose $R$. The matrix $R$ can be set as an approximate pseudoinverse matrix of $A$ (e.g., `pinv` function in Matlab/Octave). Let us move to the interval case.

**Definition 4** (Interval matrix norm)**.** For interval matrices the real matrix norm $\|\cdot\|$ can be easily generalized as

$$\|\boldsymbol{A}\| = \max\{\|A\|,\ A \in \boldsymbol{A}\}.$$

The more important implication of Theorem 3 holds also for interval matrices. The proof easily follows from the definition of interval matrix norm.

**Corollary 1.** *Let $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ be an interval matrix. Suppose there exists a real matrix $R \in \mathbb{R}^{n\times m}$ such that for an arbitrary submultiplicative matrix norm $\|\cdot\|$ the inequality*

$$\|I - R\boldsymbol{A}\| < 1 \tag{3}$$

*holds, then $\boldsymbol{A}$ has full column rank.*

The remaining task is to find the matrix $R$ and to compute the norm of an interval matrix. Inspired by the point real case, $R$ can be set as an approximate pseudoinverse of the midpoint marix of $\boldsymbol{A}$. Regarding the computation of matrix norms, there are easily computable submultiplicative matrix norms:

$$\|\boldsymbol{A}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |\boldsymbol{A}_{ij}|,$$

$$\|\boldsymbol{A}\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |\boldsymbol{A}_{ij}|.$$

We do not use the norm $\|\cdot\|_2$ here since checking whether $\|\boldsymbol{A}\|_2 < 1$ for an interval matrix $\boldsymbol{A}$ is coNP-hard even for a very specialized case [15].

However, it can happen that $\|A\|_1 \geq 1$, $\|A\|_\infty \geq 1$, even though the spectral radius $\rho(A) < 1$. Furthermore, we can use a so-called *scaled maximum norm* as a generalization of the maximum norm. For any vector $\boldsymbol{x} \in \mathbb{IR}^n$ and a vector $0 < u \in \mathbb{R}^n$ we define

$$\|\boldsymbol{x}\|_u := \max\{\ |\boldsymbol{x}_i|/u_i\ \ |\ i = 1, \dots, n\},$$

and

$$\|\boldsymbol{A}\|_u := \|\ |\boldsymbol{A}| \cdot u\ \|_u$$

which is also a submultiplicative matrix norm [16]. Let us take a look at a small example.

**Example 1.** Let us have the matrix

$$A = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.2 & 0.4 & 0.2 \\ 0.3 & 0.2 & 0.5 \end{pmatrix},$$

then $\rho(A) \approx 0.94641$, $\|A\|_1 = 1$, $\|A\|_\infty = 1$. However, for $u = (0.62, 0.45, 0.62)^T$, $\|A\|_u = 0.95111 < 1$.

The previous example showed that the scaled maximum norm can help. The question is how to choose a proper vector $u$. We know that for each $u > 0$ the spectral radius $\rho(A) \leq \|A\|_u$. The following condition can be found in [18].

$$\|A\|_u < 1 \Leftrightarrow |A| \cdot u < u. \tag{4}$$

The matrix $C = |A|$ is a non-negative matrix and hence $\rho(C) = \inf_{u>0} \|C\|_u$ [18]. Hence, to compute such $u$ we can run a few steps of the well-known power method (see e.g., [12]). It may converge to the eigenvector corresponding to the largest eigenvalue of $C$. When $\rho(C) < 1$, the approximate eigenvector might be a suitable candidate for $u$ satisfying (4).

**Algorithm 1** (finding $u$)**.** Start with some $u_0 > 0$ (possibly with $u = (1, \dots, 1)^T$). Then compute $u$ iteratively using the following formula

$$u_{k+1} = |A| u_k.$$

Algorithm stops when a certain number of steps was executed or when $|u_{k+1} - u_k| < \epsilon$ for some small positive vector $\epsilon$. Set $u = u_{k+1}$ and check the property (4).

Note that unlike the power method, it is not necessary to normalize the vectors $u_k$, since the algorithm might run only for a few steps.

**Example 2.** For the matrix from Example 1

$$A = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.2 & 0.4 & 0.2 \\ 0.3 & 0.2 & 0.5 \end{pmatrix},$$

$\rho(A) = \rho(|A|) \approx 0.94641$, let us take $u_0 = (1, 1, 1)^T$. Then

$$\begin{aligned} u_0 &= (1, 1, 1)^T, \quad \|A\|_{u_0} = 1, \\ u_1 &= (1, 0.8, 1)^T, \quad \|A\|_{u_1} = 0.96 < 1. \end{aligned}$$

**Example 3.** The algorithm can also work for signed nonsymmeric matrices. Let

$$A = \begin{pmatrix} 0.40 & -0.27 & 0.27 & 0.20 \\ 0.27 & 0.19 & 0.31 & -0.18 \\ 0.27 & -0.31 & 0.06 & 0.13 \\ 0.20 & 0.18 & 0.13 & -0.22 \end{pmatrix},$$

$\rho(A) \approx 0.306691, \rho(|A|) \approx 0.927584$, let us take $u_0 = (1,1,1,1)^T$. Then

$$\begin{aligned} u_0 &= (1,1,1,1)^T, \quad \|A\|_{u_0} = 1.14, \\ u_1 &= (1.14, 0.95, 0.77, 0.73)^T, \quad \|A\|_{u_1} = 0.96545 < 1. \end{aligned}$$

**Example 4.** This will not always help, let us have

$$A = \frac{2}{5}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

then $\rho(A) = 0.8 < \rho(|A|) = 1.6$ , let us take $u_0 = (1,1,1,1)^T$. Then

$$\begin{aligned} u_0 &= (1,1,1,1)^T, \quad \|A\|_{u_0} = 1.6, \\ u_1 &= (1.6, 1.6, 1.6, 1.6)^T, \quad \|A\|_{u_1} = 1.6 > 1. \end{aligned}$$

**Example 5.** The Algorithm may not always find a proper $u$ even though such $u$ exists. For the matrix

$$A = \begin{pmatrix} 0 & 1/2 \\ 1/2 & 1/2 \end{pmatrix},$$

$\rho(A) = \rho(|A|) \approx 0.81 < 1$. When the algorithm starts with a vector $u_0 = (1,1)^T$ then the algorithm never converges

$$\begin{aligned} u_0 &= (1,1)^T, \quad \|A\|_{u_0} = 1, \\ u_1 &= (1,1)^T, \quad \|A\|_{u_1} = 1. \end{aligned}$$

However, for $u = (0.5, 0.8)^T$ the norm $\|A\|_u = 0.8125 < 1$.

Another possibility that may help in some cases (e.g., in Example 5) is the following algorithm.

**Algorithm 2** (finding $u$)**.** Compute eigenvalues and eigenvectors of $|A|$ (in Octave/Matlab using function `eig`). Such computation is not verified. Anyway, if $\rho(|A|) < 1$ then take the corresponding eigenvector $u$ and if it is real then check in a verified way whether $\|A\|_u < 1$.

## 3.6 Relationship between the Two Sufficient Conditions

In the previous subsection the two sufficient conditions for a matrix having full column rank were introduced – (2) and (3). The question is what is the relation between these two conditions? In this section we are going to elaborate the answer to this question which is our main theoretical result of this paper.

When $\boldsymbol{A}$ is a square matrix, then both conditions are equivalent. When $A_c$ has full column rank then $A_c$ is regular and $A_c^+ = A_c^{-1}$. In such a case both conditions are equivalent to $\boldsymbol{A}$ being an H-matrix [18]. What is the relation in the rectangular case? In the following theorem we claim that the second condition is always stronger.

**Theorem 4.** *For a general matrix $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ the implication (a) ⇒ (b) holds, where*

*(a) $A_c$ has full column rank and $\rho(|A_c^+|A_\Delta) < 1$,*

*(b) $\exists u \in \mathbb{R}^n$, $u > 0$ and $\exists R \in \mathbb{R}^{n\times m}$ such that $\|I - R\boldsymbol{A}\|_u < 1$.*

*Proof.* When $A_c$ has full column rank then $A_c^+ = (A_c^T A_c)^{-1}A_c^T$, which causes the matrix $A_c^+\boldsymbol{A}$ to have the midpoint matrix equal to $I$. Hence $I - A_c^+\boldsymbol{A}$ is the matrix with the midpoint matrix 0. According to a multiplication rule for one real and one interval matrix [18], it has the radius matrix equal to $|A_c^+|A_\Delta$. Therefore, for all $C \in I - A_c^+\boldsymbol{A}$ it holds that $|C| \leq |A_c^+|A_\Delta$. It is well-known that for matrices $C, D$ the following holds

$$|C| < D \Rightarrow \rho(C) < \rho(D). \tag{5}$$

Together with (a) it gives

$$\rho(C) \leq \rho\left(|A_c^+|A_\Delta\right) < 1.$$

By [18], there must exist some $u > 0$ such that $\|C\|_u < 1$ for each $C \in I - (A_c)^+\boldsymbol{A}$. According to the definition of scaled maximum norm there must exist $u > 0$ such that $\|I - A_c^+\boldsymbol{A}\|_u < 1$. Finally, to make *(b)* hold, set $R = (A_c^T A_c)^{-1}A_c^T$. □

Before we shed more light on the other implication, we prove the following result. It shows that using the Moore-pseudoinverse preconditioner is optimal from some point of view and under specific assumptions. The proof is an adaptation of the analogous proof for square systems [20].

**Theorem 5.** *Assume that $R \in \mathbb{R}^{n\times m}$ has the form of $R = CA_c^+$, where $C \in \mathbb{R}^{n\times n}$ is nonsingular. If*

$$\rho(|I - RA_c| + |R|A_\Delta) < 1, \tag{6}$$

*then $A_c$ has full column rank and*

$$\rho(|A_c^+|A_\Delta) \leq \rho(|I - RA_c| + |R|A_\Delta).$$

*Proof.* We have

$$\rho(I - RA_c) \leq \rho(|I - RA_c|) \leq \rho(|I - RA_c| + |R|A_\Delta) < 1. \tag{7}$$

Thus, $RA_c$ is nonsingular and $A_c$ has full column rank. Again, in this case $A_c^+ = ({A_c}^T A_c)^{-1}A_c^T$ and $A_c^+A_c = I$.

Now, define

$$G := |I - RA_c| + |R|A_\Delta + \epsilon ee^T, \quad \alpha := \rho(G) < 1,$$

where $\epsilon > 0$ is small enough. Such $\epsilon$ exists due to continuity of the spectral radius [8, 12]. Since $G > 0$, by Perron–Frobenius Theorem there exists $0 < x \in \mathbb{R}^n$ such that $Gx = \alpha x$. Using the fact that $\alpha < 1$, we derive

$$\alpha|I - RA_c|x \le |I - RA_c|x \le |I - RA_c|x + |R|A_\Delta x < \alpha x,$$

and when we combine the last and then the first inequality we get

$$|R|A_\Delta x < \alpha(I - |I - RA_c|)x.$$

By (7) and Neumann series theory, $I - |I - RA_c|$ has a nonnegative inverse, which yields

$$(I - |I - RA_c|)^{-1}|R|A_\Delta x < \alpha x.$$

Now, from

$$\begin{aligned} A_c^+ &= (CI)^{-1}CA_c^+ = (CA_c^+A_c)^{-1}CA_c^+ = (RA_c)^{-1}R \\ &= (I - (I - RA_c))^{-1}R = \sum_{i=1}^{\infty}(I - RA_c)^iR \end{aligned}$$

we derive

$$|A_c^+| \le \sum_{i=1}^{\infty}|I - RA_c|^i|R| = (I - |I - RA_c|)^{-1}|R|.$$

Putting all together, we obtain

$$|A_c^+|A_\Delta x \le (I - |I - RA_c|)^{-1}|R|A_\Delta x < \alpha x.$$

By Perron–Frobenius theory, $\rho(|A_c^+|A_\Delta) < \alpha < 1$, from which the statement follows having $\epsilon \to 0$ due to continuity of the spectral radius. □

Even though the assumption on $R$ of the above theorem is quite restrictive, it covers a lot of natural choices: Not only the pseudoinverse $R = A_c^+$, but also $R = A_c^T$ and their multiples, among others.

The following example shows that $A_c^+$ is not the best preconditioner in general.

**Example 6.** Let

$$A_c = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \quad A_\Delta = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} -1.5385 & 0.0769 & 0.4615 \\ 1.2404 & 0.0192 & -0.2596 \end{pmatrix}.$$

Then $\rho(|A_c^+|A_\Delta) = 1.0417 > 1$, so full column rank of $\boldsymbol{A}$ is not confirmed yet. However, using the sufficient condition for regularity of $R\boldsymbol{A}$ in [21], we get

$$\rho(|I_n - RA_c| + |R|A_\Delta) = 0.8990 < 1,$$

confirming full column rank of $\boldsymbol{A}$.

Using the previous theorem we can formulate the other implication between the two sufficient conditions. However, we need to modify the second condition a little.

**Theorem 6.** *For a general matrix $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ the implication (a) $\Leftarrow$ (b*) holds, where*

*(a)* *$A_c$ has full column rank and $\rho(|(A^c)^+|A^\Delta) < 1$,*

*(b*)* *$\exists u \in \mathbb{R}^n, u > 0$, $\exists R = (CA_c^+) \in \mathbb{R}^{n\times m}$, for some nonsingular $C \in \mathbb{R}^{n\times n}$ such that $\|I - R\boldsymbol{A}\|_u < 1$.*

*Proof.* The statement (b*) is equal to $|I - R\boldsymbol{A}|u < u$. The midpoint matrix of $I - R\boldsymbol{A}$ is $I - RA_c$ and the radius matrix is $|R|A_\Delta$. Hence,

$$|I - R\boldsymbol{A}| = |I - RA_c| + |R|A_\Delta$$

(see e.g., [18]) and $(|I - RA_c| + |R|A_\Delta)u < u$. Also by [18], because the whole matrix on the left side is nonnegative, the formula is equivalent to $\rho(|I - RA_c| + |R|A_\Delta) < 1$ and according to Theorem 5 the claim *(a)* holds. □

# 4 Solvability

In the final comparison of the mentioned methods for detecting unsolvability a method for detecting the opposite case – solvability of a system – might bring a new information to understanding the bigger picture. That is why this small section is devoted to this topic. In the following, unsolvability is understood as the opposite of weak solvability defined in [22].

**Definition 5** (Weak solvability)**.** An interval system $\boldsymbol{A}x = \boldsymbol{b}$ is called *(weakly) solvable* if there exists a vector $x$ such that $Ax = b$ holds for some $A \in \boldsymbol{A}, b \in \boldsymbol{b}$.

In another words, an interval system is solvable if there exists a solvable real system contained in it. Since detecting unsolvability is a coNP-complete problem, detecting solvability is NP-complete. That is why we focus only on sufficient conditions here. For example, consider the midpoint system $A_cx = b_c$. This system is possibly unsolvable, that is why we set

$$x = A_c^+ b_c.$$

The vector $x$ may not be a solution of the midpoint system, however, we assume that $x$ is a solution of a system that is close enough to the midpoint system, and hence still contained in the original interval system. We can check this by applying Theorem 1 on $x$. The checking must be done in a verified way using interval arithmetics.

Furthermore, the vector $\mathrm{sgn}(x)$ gives us a hint in which orthant the solution can be found. With such knowledge we can rewrite the Oettli-Prager formula for the given orthant and apply verified linear programming.

# 5 Comparison of the Methods

In this section we compare the previously discussed methods for detecting unsolvability; namely:

- `ge` – the Gaussian elimination approach described in Section 3.2,
- `subsq` – the square subsystems approach described in Section 3.3, with 5 random square subsystem choices,
- `lsq` – the least squares approach discussed in Section 3.4,

- `fcr` – the approach using full column rank sufficient condition (3) with maximum norm described in Section 3.5,
- `fcrit` – the approach using the condition (3), with scaled maximum norm and iterative search for a vector $u$ described in Section 3.5,
- `eig` – the approach using condition (2) with non-verified computation of spectral radius described in Section 3.5.

The method `eig` is shown for comparison purpose only, it is not a verified method since the spectral radius in the formula is not computed in a verified way. Those methods that need to compute an enclosure of a square interval system use the method "\" from the Octave interval package.

The methods are tested on random systems with intervals having fixed radius. The radius range is selected to suit a particular method – to catch the region of its applicability. The methods were applied to systems with various number of variables ($n = 5, 10, 15, \ldots, 100$), the number of equations $m$ was always selected according to $n$ as $m = \frac{3}{2}n$ to form a rectangular system. For each combination of a radius and a system size, 100 random systems were generated and tested for unsolvability by various methods.

The generation of random systems works as follows. First, a midpoint matrix $A_c$ and a midpoint right-hand side $b_c$ is generated using uniformly random numbers drawn from the interval $[-1, 1]$. The generated system $A_c x = b_c$ is most likely to be unsolvable. Then, its each coefficient is enclosed with an interval having a prescribed fixed radius.

The results of testing are displayed in Figure 2. A point on a heatmap shows the percentage of systems that were detected to be unsolvable for a given method (the label above the heatmap), a given system size ($x$-axis) and a given radius ($y$-axis). Note that, even though, the sizes ($x$-axes) remain the same, the interval radii range ($y$-axes) might change from method to method. There are basically two types of methods. The first group works only for "smaller" radii relative to the coefficients of $A_c, b_c$ ($r < 0.01$) and "smaller" system sizes ($n < 40$) – `ge`, `lsq`, `subsq`. The methods in the second group work even for "larger" radii ($r < 1$) – `fcr`, `fcrit`, `eig`.

The method `ge` works only for very small systems. Since for detection of unsolvability it must be used without preconditioning, the interval operations cause large overestimation that will occur for larger systems ($n > 10$) and the Gaussian elimination will find a solution or it will not be able to proceed because all pivot intervals contain 0 at some step.

The methods `lsq` and `subsq` detect unsolvability with a similar accuracy. The efficiency and the computation time of `subsq` depend on the number of random square subsystems inspected. Both methods depend on the efficiency of a method used for computing enclosures of square interval systems.

The absolute winners are the methods `fcr` and `fcrit`. The `frc` is the fastest method (the computation for a $100 \times 100$ system took approximately 0.473 seconds on a laptop with Intel Core i5-7200U – 2.5GHz, TB 3.1GHz, HyperThreading; 8GB DDR4 memory). In the tested cases, the iterative search for scaled maximum norm seemed to help. It adds only some minor computational time, the computation took approximately 0.476 seconds.

The method `eig` returned great results too, however because of non-verified computation it did not return verified results and therefore it was excluded from the competition. Nevertheless, the heatmaps of `eig` and `fcrit` look very similar. Since in Theorem 4 we proved that the maximum scaled norm condition is stronger, it is not a surprise. The strength of the scaled maximum norm condition stands and falls on finding a proper vector $u$. In this case, the heatmaps show, that our heuristic iterative search for $u$ does the job very well.

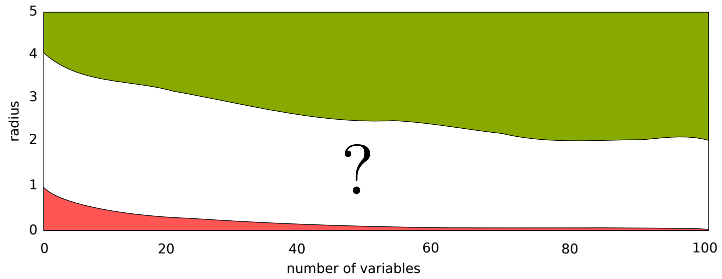

Figure 1: Detection gap between unsolvable (red) and solvable (green) systems. The borders occured by approximate interpolation of the 50% level of the corresponding heatmaps.

With growth of interval widths, generated systems become solvable. For detecting this we applied the previously described condition for detecting solvability to each system. In Figure 2 the solvability heatmap is also included. We would like to point out the detection gap between unsolvability and solvability. It is displayed in Figure 1. The white area marked by questionmark forms a band of systems of which neither unsolvability nor solvability is decidable by the mentioned methods. Since the detection of solvability and unsovability are NP-complete and coNP-complete problems respectively, the presence of such gap is no surprise. We can only try to further reduce such gap unless P≠NP.

## 6 Conclusion

In this work we discussed various methods for detecting unsolvability of interval linear systems. Many of those methods or sufficient conditions were previously known, however, used for a different purpose. We showed how to modify or improve them to detect unsolvability. The methods were tested on random rectangular systems of size up to 100 variables. The results were displayed using heatmaps which brought a new interesting insight. According to the heatmaps the best methods use the two sufficient conditions for a matrix having full column rank. The heatmaps posted an interesting question about the relation of these two conditions. We proved that under a certain assumption these conditions are equivalent. The first condition using scaled maximum norm proved to be more computably feasible, however it might be a problem to find a proper scaled maximum norm. We showed two heuristic algorithms for finding such a norm, however they do not work universally. That is why the further work might concern more efficient ways of finding a proper scaled maximum norm. In the last section we pointed out the unsolvability gap - the ”no man’s land” of sufficient unsolvability and solvability polynomially checkable conditions. The further research task might be to bring the borders of such gap closer to each other.

## References

[1] Helmut Beeck. *Linear programming with inexact data.* Technische Universität München. Institut für Statistik und Unternehmensforschung, 1978.

[2] Julien Alexandre dit Sandretto and Milan Hladík. Solving over-constrained systems of non-linear interval equations–and its robotic application. *Applied Mathematics and Computation*, 313:180–195, 2017.

[3] Eldon Hansen and G William Walster. Solving overdetermined systems of interval linear equations. *Reliable computing*, 12(3):239–243, 2006.

[4] Milan Hladík. New operator and method for solving real preconditioned interval linear equations. *SIAM J. Numer. Anal.*, 52(1):194–206, 2014.

[5] Jaroslav Horáček and Milan Hladík. Computing enclosures of overdetermined interval linear systems. *Reliable Computing*, 19(2):142–155, 2013.

[6] Jaroslav Horáček and Milan Hladík. Subsquares approach–a simple scheme for solving overdetermined interval linear systems. In *International Conference on Parallel Processing and Applied Mathematics*, pages 613–622. Springer, 2013.

[7] Jaroslav Horáček, Milan Hladík, and Michal Černý. Interval linear algebra and computational complexity. In *International Conference on Matrix Analysis and its Applications*, pages 37–66. Springer, 2015.

[8] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

[9] Christian Keil. Lurupa-rigorous error bounds in linear programming. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.

[10] Vladik Kreinovich, Anatoly Lakeyev, Jiří Rohn, and Patrick Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer, Dordrecht, 1998.

[11] Günter Mayer. *Interval analysis: and automatic result verification*, volume 65. Walter de Gruyter GmbH & Co KG, 2017.

[12] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 71. Siam, 2000.

[13] J Minisini, A Rauh, and EP Hofer. Carleman linearization for approximate solutions of nonlinear control problems: Part 1–theory. In *Proc. of the 14th Intl. Workshop on Dynamics and Control*, pages 215–222, 2007.

[14] Ramon E. Moore, R Baker Kearfott, and Michael J. Cloud. *Introduction to Interval Analysis*. SIAM, 2009.

[15] Arkadii Nemirovskii. Several np-hard problems arising in robust stability analysis. *Mathematics of Control, Signals and Systems*, 6(2):99–105, 1993.

[16] Arnold Neumaier. New techniques for the analysis of linear interval equations. *Linear Algebra and its Applications*, 58:273–325, 1984.

[17] Arnold Neumaier. Linear interval equations. In *Interval Mathematics 1985*, pages 109–120. Springer, 1986.

[18] Arnold Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.

[19] Andreas Rauh, Johanna Minisini, and Eberhard P Hofer. Towards the development of an interval arithmetic environment for validated computer-aided design and verification of systems in control engineering. In *Numerical Validation in Current Hardware Architectures*, pages 175–188. Springer, 2009.

[20] Georg Rex and Jiří Rohn. A note on checking regularity of interval matrices. *Linear and Multilinear Algebra*, 39(3):259–262, 1995.

[21] Georg Rex and Jiří Rohn. Sufficient conditions for regularity and singularity of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(2):437–445, 1998.

[22] Jiří Rohn. Solvability of systems of interval linear equations and inequalities. In *Linear optimization problems with inexact data*, pages 35–77. Springer, 2006.

[23] Jiří Rohn. Verification of linear (in)dependence in finite precision arithmetic. *Mathematics in Computer Science*, 8(3–4):323–328, 2014.

[24] Jiří Rohn. A handbook of results on interval linear problems. Technical Report 1163, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2012. Available at http://www.nsc.ru/interval/Library/InteBooks/!handbook.pdf.

[25] Siegfried M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta*

*Numer.*, 19:287–449, 2010.

[26] Sergey Petrovich Shary. On full-rank interval matrices. *Numerical Analysis and Applications*, 7(3):241–254, 2014.

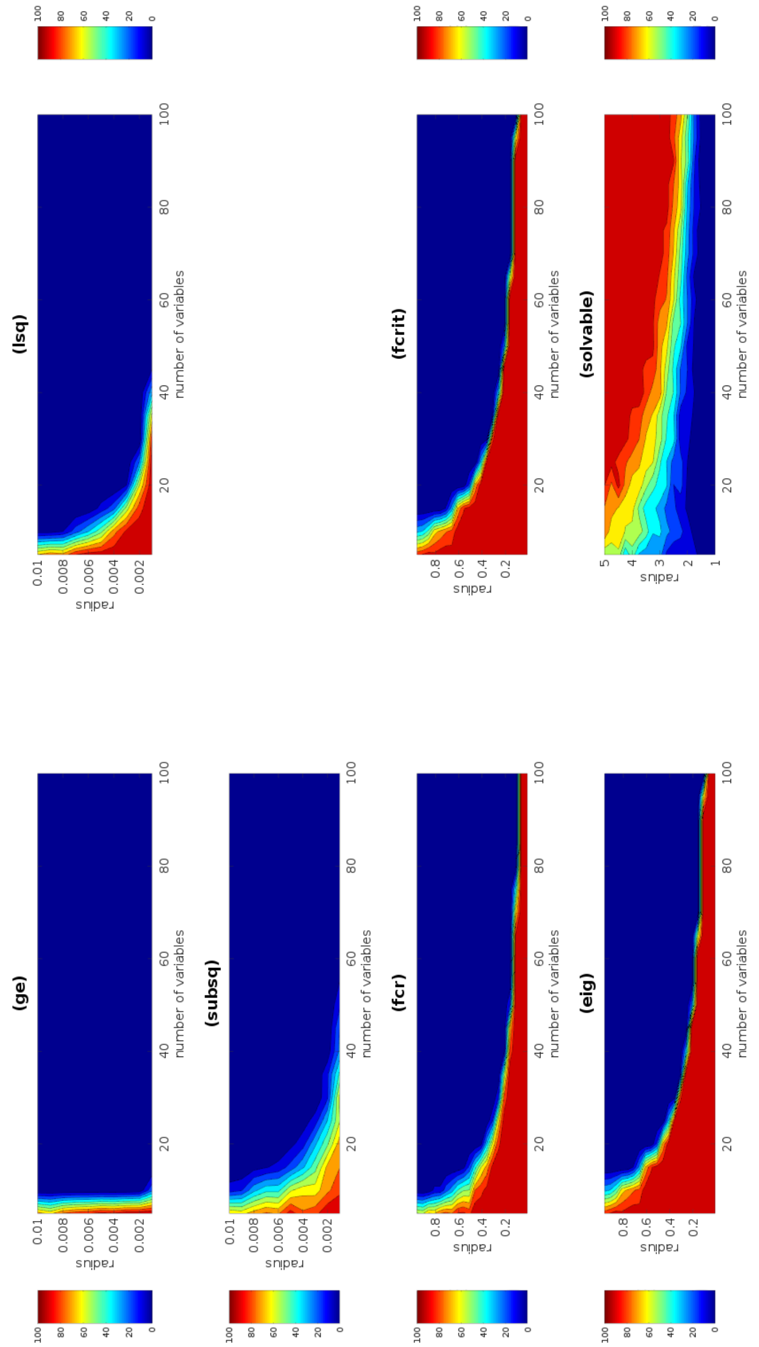

Figure 2: Comparison of methods for unsolvability and solvability detection. The horizontal axis shows the sizes of systems ($n$ - number of variables), the vertical axis shows radii of system coefficients (note that it may differ for various heatmaps). The color (heat) shows the percentage of succesfull detection based on attached legend. Each heatmap corresponds to a methods specified by a label above the heatmap. The (solvable) heatmap corresponds to the solvability detection condition.