Computer Assisted Mechanics and Engineering Sciences, **5**: ??-??, 1998.

Analysis of linear mechanical structures with uncertainties by means of interval methods

Zenon Kulpa
Institute of Fundamental Technological Research, Polish Academy of Sciences, ul. Öwiëtokrzyska 21, 00-049 Warsaw, Poland

Andrzej Pownuk
Chair of Theoretical Mechanics, Silesian Technical University, ul. Krzywoustego 7, 44-100 Gliwice, Poland

Iwona Skalna
Department of Management, Academy of Mining and Metallurgy, ul. Gramatyka 10, 30-067 Cracow, Poland

(Received December ??, 1997)

One of the simplest ways of representation of uncertain or inexact data, as well as inexact computations with them, is based on *interval arithmetic*. In this approach, an uncertain (real) number is represented by an *interval* (a continuous bounded subset) of real numbers which presumably contains the unknown exact value of the number in question. Despite its simplicity, it conforms very well to many practical situations, like tolerance handling or managing rounding errors in numerical computations. Also, the so-called *α-cut* method of handling fuzzy sets membership functions is based on replacing a fuzzy set problem with a set of interval problems.

The purpose of this paper is to investigate possibilities of and problems with application of interval methods in (qualitative) analysis of linear mechanical systems with parameter uncertainties, in particular truss structures and frames. The paper starts with an introduction to interval arithmetic and systems of linear interval equations, including an overview of basic methods for finding interval estimates for the set of solutions of such systems. The methods are further illustrated by several examples of practical problems, solved by our hybrid system of analysis of mechanical structures. Finally, several general problems with using interval methods for analysis of such linear systems are identified, with promising avenues for further research indicated as a result. The problems discussed include estimation inaccuracy of the algorithms (especially the fundamental problem of matrix coefficient dependence), their computational complexity, as well as inadequate development of methods for analysis of interval systems with singular matrices.

1. INTRODUCTION

Qualitative analysis is an area of AI research attempting to model the everyday, qualitative, non-numerical reasoning humans use to estimate the range of possible solutions to some real-world problems, especially in the case of inexact or incomplete data [6, 7, 11, 13]. Among many problems considered in this field is the problem of representing, more or less qualitatively, uncertain or inexact data. One of the simplest ways of representation of such data, as well as inexact computations on them, is based on the so-called *interval arithmetic* [1, 19], called also *interval analysis*. In this approach, an uncertain (real) number is represented by an *interval* (a continuous bounded subset) of real numbers which presumably contains the unknown exact value of the number in question. Thus, the uncertainty is bounded by the size of the interval; in addition, no commitment to a particular probability distribution (or its estimate) of the of various alternative values within the interval needs to be made. This model is thus in principle much simpler than general probabilistic or fuzzy set formulations of uncertainty. Despite its simplicity, it conforms very well to many practical situations, like tolerance handling in mechanics or managing

rounding errors in numerical computations. Hence the number of applications of interval-based numerical methods is growing steadily.

Also, the so recently popular *fuzzy set* approach uses more and more often the interval arithmetic formulations and methods. First, intervals can be considered as a special kind of *fuzzy set membership function* (a "square-wave" membership function). Second, the so-called α*-cut* approach [3] to handling of membership function shapes is based on replacing a fuzzy set problem with a set of interval problems: every interval problem is obtained by thresholding the original fuzzy set membership function at some value α of the function, $0 \leq \alpha \leq 1$. It is thus not surprising (though possibly a little annoying for interval analysis researchers) that quite often a paper devoted, according to its title, to fuzzy set methods may deal mostly with interval analysis inside [23, 34].

As far as the authors know, the first work on interval arithmetic appeared in 1956 in a Polish mathematical journal, authored by Warmus [35]. It contained most of the basic notions and definitions for interval arithmetic, including some its extensions (the one now known as "*Kaucher arithmetic*", see the footnote in Sec. 2.1). The field reached a sort of maturity in middle sixties with the publication by Moore [19] of the first book on the subject. Another basic textbook appeared in 1974 (in German; an English translation, revised and expanded, appeared in 1983 [1]). Since then, most applications of interval analysis concerned the problem of assuring reliable computations in the presence of rounding errors in computer arithmetic[1)]; recently they also found significant applications in the field of optimisation [8]. Basic interval arithmetic techniques used for solving systems of linear and non-linear equations are covered in [20]. Experiences with application of interval methods to analysis of some mechanical systems have been reported in several papers, among others [12, 23, 25, 34].

The purpose of the work described in this paper was to investigate possibilities of and problems with application of interval methods in (qualitative) analysis of linear mechanical systems with parameter uncertainties, in particular truss structures and frames. The paper starts with an introduction to interval arithmetic and systems of linear interval equations. Then follows an overview of basic methods for finding interval estimates for the set of solutions of such systems. It includes methods giving exact estimates (though computationally inefficient) as well as faster methods producing approximate estimates. The usefulness of the methods for practical problems is further evaluated on several examples of solving linear mechanical problems with uncertain parameters (trusses and frames), as implemented in our hybrid system of analysis of mechanical structures [16, 17]. Finally, several general problems with using interval methods for analysis of such linear systems are identified. Most of all, the estimation inaccuracy of the algorithms, especially due to the problem of matrix coefficient dependence (in turn the result of certain fundamental algebraical weaknesses of the interval arithmetic), are discussed in some detail. The problems of computational complexity of the algorithms as well as inadequate development of methods for analysis of singular interval systems are also discussed. The discussion leads to formulation of several urgent and promising avenues for further research.

2. INTERVAL ARITHMETIC: BASIC NOTIONS

The interval notation used in the paper follows basically that of Neumaier [20], with minor modifications and some additions. Modifications consist mostly of using an operator instead of functional notation for certain basic interval functions in order to minimise the number of superfluous parentheses in formulas, as well as using boldface font for vectors and matrices, as is customary in computational mechanics circles.

2.1. Real intervals

In general, an *interval* is defined as a *pair of elements* of some (at least *partially*) *ordered set* [14, 15]. For our purposes here, and in accord with standard treatment of intervals in interval arithmetic [1, 20], we

[1)] Spectacular examples of such errors are described e.g. in [8, 31]. The two recent technical disasters caused by computer arithmetic errors are the *Patriot* missile failure in Dharan, Saudi Arabia, in 1991, and the explosion during launch of the *Ariane 5* rocket in Kourou, French Guyana, in 1996 (see `http://www.math.psu.edu/dna/455.f96/disasters.html`).

shall identify an interval with the set of elements lying between the interval endpoints (including the endpoints, hence all intervals are *closed* sets), and using the set of real numbers as the underlying ordered set (hence restricting ourselves to so-called *real intervals*). Thus, a (*proper*) *real interval* x is a subset of the set of real numbers R such that:

$$x = [\underline{x}, \overline{x}] = \{\tilde{x} \in \mathsf{R} \mid \underline{x} \le \tilde{x} \le \overline{x}\},$$

where $\underline{x} \le \overline{x}$ and $\underline{x} = \sup x$, $\overline{x} = \inf x$ are *endpoints* of the interval x. In general, by $\tilde{x}$ we shall denote any element of the interval x. The set of all such intervals is denoted by $\mathbb{IR}$ and called a (*real*) *interval space*[2)]. The interval is called *thick* if $\underline{x} < \overline{x}$; *thin* (or *point*) interval if $\underline{x} = \overline{x}$. Thin intervals for most purposes can be identified with corresponding real numbers.

For a real interval x the *midpoint*, *radius*, *magnitude* (or *absolute value*) and *mignitude* of the interval are defined, respectively, as follows:

$$\breve{x} = \mathrm{mid}\ x = (\overline{x} + \underline{x})/2,$$

$$\mathrm{rad}\ x = (\overline{x} - \underline{x})/2,$$

$$|x| = \mathrm{mag}\ x = \max(|\underline{x}|, |\overline{x}|),$$

$$\langle x \rangle = \mathrm{mig}\ x = \min(|\underline{x}|, |\overline{x}|) \ \text{ if } \ 0 \notin x, \ \langle x \rangle = 0 \ \text{ otherwise.}$$

Often it is convenient to consider also the width of an interval, defined as:

$$\mathrm{wid}\ x = \overline{x} - \underline{x} = 2\,\mathrm{rad}\ x.$$

Since $x = [\breve{x} - \mathrm{rad}\ x, \breve{x} + \mathrm{rad}\ x]$, intervals can be also expressed in terms of the midpoint and radius (instead of endpoints)—a so-called *centered formulation* (first introduced by Warmus [35], see also the centered diagrammatic representation for a space of intervals in [14, 15]).

As intervals are considered here to be sets of reals (or of n-tuples of reals, see below), a set-theoretic operations and relations on intervals can be used, in particular:

$$x \cap y = \left\{\tilde{x} \in \mathsf{R} \mid \tilde{x} \in x \ \wedge \ \tilde{x} \in y\right\} = \begin{cases} \varnothing & \text{if } \ \overline{x} < \underline{y} \ \text{ or } \ \overline{y} < \underline{x}, \\ [\max(|\underline{x}|, |\underline{y}|), \min(|\overline{x}|, |\overline{y}|)] & \text{otherwise.} \end{cases}$$

$$x \subseteq y \Leftrightarrow \underline{y} \le \underline{x} \ \wedge \ \overline{x} \le \overline{y}.$$

Also, the notions of *interior* and *boundary* of an interval are sometimes useful. For thick intervals they are defined as:

$$\mathrm{int}\ x = \{\tilde{x} \in \mathsf{R} \mid \underline{x} < \tilde{x} < \overline{x}\},$$

$$\partial x = \{\underline{x}, \overline{x}\}.$$

For thin intervals, in a somewhat non-standard way, interior and boundary are identified with the number x itself, so that $\mathrm{int}\,[x, x] = \partial[x, x] = x$.

Operations and functions on reals are naturally extended to interval operands according to the general formula:

$$\begin{aligned} x\ \mathbf{op}\ y &= \left\{\tilde{x}\ \mathbf{op}\ \tilde{y} \mid \tilde{x} \in x \ \ \& \ \ \tilde{y} \in y\right\}, \\ \mathrm{f}(x_1, x_2, \ldots, x_n) &= \left\{\mathrm{f}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \mid (\forall i = 1, 2, \ldots, n)\ \tilde{x}_i \in x_i\right\}. \end{aligned} \tag{1}$$

For many operations, including standard arithmetic operations of *addition*, *subtraction*, *multiplication* and *division*, the resulting set is also an interval that can be conveniently defined in terms of endpoints of the argument intervals:

[2)] Extensions of interval arithmetic that allow *improper* intervals (for which $\underline{x} > \overline{x}$) are also considered and used in practical applications (two main such extensions are called *Kaucher* and *Kahan* arithmetic, respectively, see e.g. [1, 18, 32]), but they will not be discussed here in detail.

$$
\begin{aligned}
&x + y = [\underline{x} + \underline{y},\ \overline{x} + \overline{y}],\\
&x - y = [\underline{x} - \overline{y},\ \overline{x} - \underline{y}],\\
&x \cdot y = [\min(\underline{x}\,\underline{y},\ \overline{x}\,\underline{y},\ \underline{x}\,\overline{y},\ \overline{x}\,\overline{y}),\ \max(\underline{x}\,\underline{y},\ \overline{x}\,\underline{y},\ \underline{x}\,\overline{y},\ \overline{x}\,\overline{y})],\\
&x / y = x \cdot [1/\overline{y},\ 1/\underline{y}] \ \text{ if } \ 0 \notin y, \text{ undefined otherwise}^{3)}.
\end{aligned}
\tag{2}
$$

As can be seen from the definitions, subtraction and division of real intervals are not the inverse operations to addition and multiplication, respectively, differently than for the corresponding operations on reals. Other differences exist, most notably the *distributive law* $a\,(b + c) = ab + ac$ does not hold in general in interval arithmetic. Instead, we have the weaker *subdistributive law*:

$$x\,(y + z) \subseteq xy + xz, \ \text{ for } x, y, z \in \mathbb{IR}. \tag{3}$$

These properties of interval arithmetic lead to problems with proper transformation and calculation of numerical values for interval expressions. One should exercise proper caution with applying the usual arithmetic rules and formulas like Eq. (2) to interval expressions; for correct result, the conclusive rule is always that of Eq. (1). The problem is further discussed in detail in Sec. 5.2.

Fortunately, due to another important property of arithmetic operations on intervals called *inclusion isotonicity*:

$$x \subseteq y \ \&\ u \subseteq v \ \Rightarrow\ x \ \mathbf{op}\ u \ \subseteq\ y \ \mathbf{op}\ v, \ \text{ for } x, y, u, v \in \mathbb{IR} \text{ (if } y \ \mathbf{op}\ v \text{ is defined)}, \tag{4}$$

the result of a straightforward calculation of interval expression will always include the proper result (i.e., we get at most an overestimation of the proper resulting interval).

For any bounded set of real numbers s we can define a *smallest interval enclosure* of the set, called also (interval) *hull* of the set:

$$\operatorname{hull} s = [\inf s,\ \sup s].$$

For instance, for a two-element set $\{a, b\}$ of real numbers we have $\operatorname{hull}\{a, b\} = [\min(a, b), \max(a, b)]$. Of course, if x is an interval, then $\operatorname{hull} x = \operatorname{hull} \partial x = x$.

When the application of Eq. (1) for some function or operation produces a set which is not an interval, the hull of the set can be taken if there is a need to stay within interval arithmetic all the time (which is usually the case). Hence, Eq. (1) is usually used in the form:

$$
\begin{aligned}
&x \ \mathbf{op}\ y = \operatorname{hull}\left\{\tilde{x} \ \mathbf{op}\ \tilde{y} \,\middle|\, \tilde{x} \in x \ \ \&\ \ \tilde{y} \in y\right\},\\
&\mathrm{f}(x_1, x_2, \ldots, x_n) = \operatorname{hull}\left\{\mathrm{f}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \,\middle|\, (\forall i = 1, 2, \ldots, n)\ \tilde{x}_i \in x_i\right\}.
\end{aligned}
\tag{1′}
$$

It is easy to see that the inclusion isotonicity rule of Eq. (4) is still valid with this definition of interval operations.

2.2. Interval vectors and matrices

Vectors and matrices whose elements are intervals (possibly some of them thin) are called *interval vectors* and *interval matrices*, respectively. As vectors are, in a sense, special cases of matrices, most of the discussion concerning matrices in the sequel applies equally to vectors.

Most operations on intervals can be extended to interval matrices, by applying them componentwise to all matrix elements. In particular, *infimum*, *supremum*, *midpoint*, *radius*, *magnitude*, *intersection* and *inclusion*, with corresponding notation, are so defined, as are *addition* and *subtraction*. Hence, an interval matrix $\mathbf{A} \in \mathbb{IR}^{n\times m}$ can be also considered as a set of real matrices $\tilde{\mathbf{A}}$, i.e. we can write $\tilde{\mathbf{A}} \in \mathbf{A}$ when $\underline{\mathbf{A}} \le \tilde{\mathbf{A}} \le \overline{\mathbf{A}}$, or as a *matrix interval*: $\mathbf{A} = [\underline{\mathbf{A}}, \overline{\mathbf{A}}] = [\check{\mathbf{A}} - \operatorname{rad}\mathbf{A},\ \check{\mathbf{A}} + \operatorname{rad}\mathbf{A}]$. Matrix multiplication is defined like for real matrices. Alternatively, Eq. (1′) should be used, since the set

[3] In *Kahan arithmetic* (see the previous footnote), division is always defined.

$\{\tilde{\mathbf{A}}\tilde{\mathbf{B}} \mid \tilde{\mathbf{A}} \in \mathbf{A} \ \& \ \tilde{\mathbf{B}} \in \mathbf{B}\}$ in general may not be an interval matrix. It is also important to remember that multiplication of interval matrices, (contrary to the non-interval matrices and scalar intervals), is not associative, thus in general $\mathbf{A}(\mathbf{BC}) \neq (\mathbf{AB})\mathbf{C}$, unless $\mathbf{A}$ and $\mathbf{C}$ are thin (i.e., real) matrices.

The *boundary* of an interval matrix consists of 2^t elements, where t is the number of thick interval elements of $\mathbf{A}$, and is defined as:

$$\partial\mathbf{A} = \{\tilde{\mathbf{A}} \in \mathbf{A} \mid \tilde{a}_{ij} \in \partial a_{ij}\} = \{\tilde{\mathbf{A}} \in \mathbf{A} \mid \tilde{a}_{ij} = \underline{a}_{ij} \vee \tilde{a}_{ij} = \overline{a}_{ij}\}. \tag{5}$$

Obviously, $\underline{\mathbf{A}}, \overline{\mathbf{A}} \in \partial\mathbf{A}$. The *interior* is defined straightforwardly.

Somewhat confusingly (following Neumaier [20]), we shall use the notation $\langle\mathbf{A}\rangle$ not for a componentwise mignitude of $\mathbf{A}$, but for a (real) matrix with coefficients defined by:

$\langle\mathbf{A}\rangle_{ii} = \langle a_{ii}\rangle$,

$\langle\mathbf{A}\rangle_{ik} = -|a_{ik}|$, for $i \neq k$.

Such defined matrix (called a *comparison matrix*) has important uses in analysis of linear systems of equations, both real and interval. Also, the so called *M-matrices* and *H-matrices* are of importance here:

- A square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is an *M-matrix* when $a_{ik} \leq 0$ for all $i \neq k$, and $\mathbf{Au} > 0$ for some positive vector $\mathbf{u} \in \mathrm{R}^n$.
- An *H-matrix* is a matrix $\mathbf{A}$ whose comparison matrix $\langle\mathbf{A}\rangle$ is an *M*-matrix, which is equivalent to the condition $\langle\mathbf{A}\rangle\mathbf{u} > 0$ for some positive vector $\mathbf{u} \in \mathrm{R}^n$.

Every *M*-matrix is an *H*-matrix.

For the given real positive (scaling) vector $\mathbf{u} \in \mathrm{R}^n$ we also define the *scaled maximum norm* (or *scaled row sum norm*) of an interval matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ as follows:

$$\|\mathbf{A}\|_{\mathbf{u}} = \max_{i=1,\ldots,n} \sum_{k=1}^{n} |a_{ik}| u_k / u_i.$$

For $\mathbf{u} = (1, 1, \ldots, 1)^{\mathrm{T}}$ this reduces to the standard maximal row sum norm $\|\mathbf{A}\|_{\infty}$.

A square interval matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called *regular* (or *non-singular*) if all real matrices $\tilde{\mathbf{A}} \in \mathbf{A}$ are regular (non-singular); otherwise it is called *singular*. *H*-matrices (hence *M*-matrices) are all regular. A matrix is called *strongly regular* if the so-called *preconditioned matrix* $\check{\mathbf{A}}^{-1}\mathbf{A}$ is regular. There are several other equivalent conditions of strong regularity [20], e.g. that $\check{\mathbf{A}}^{-1}\mathbf{A}$ is an *H*-matrix, or that $\rho(|\check{\mathbf{A}}^{-1}| \, \mathrm{rad}\, \mathbf{A}) < 1$ (where $\rho(\mathbf{A})$ is a spectral radius of $\mathbf{A}$), or that $\|\mathbf{I} - \check{\mathbf{A}}^{-1}\mathbf{A}\|_{\mathbf{u}} < 1$ for some $\mathbf{u} > 0$.

An *inverse* $\mathbf{A}^{-1}$ of a regular interval matrix $\mathbf{A}$ is defined as:

$$\mathbf{A}^{-1} = \mathrm{hull}\,\{\tilde{\mathbf{A}}^{-1} \in \mathrm{R}^{n\times n} \mid \tilde{\mathbf{A}} \in \mathbf{A}\}.$$

It is important to note that usually the set of inverses of matrices belonging to $\mathbf{A}$ is not an interval matrix, hence taking of the hull in the above formula is in general necessary.

If the matrix $\mathbf{A}$ is an *M*-matrix, or $\underline{\mathbf{A}}, \overline{\mathbf{A}}$ are regular and $\underline{\mathbf{A}}^{-1}, \overline{\mathbf{A}}^{-1} \geq 0$, we have simply $\mathbf{A}^{-1} = [\overline{\mathbf{A}}^{-1}, \underline{\mathbf{A}}^{-1}] \geq 0$. Interval matrices which are regular and for which $\mathbf{A}^{-1} \geq 0$ are called *inverse positive*. *M*-matrices are inverse positive, but *H*-matrices in general are not.

3. SYSTEMS OF LINEAR INTERVAL EQUATIONS

Let us consider a linear interval system of equations with an interval coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and an interval right-hand vector $\mathbf{b} \in \mathbb{R}^n$:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{6}$$

The solution set of Eq. (6) is usually defined as:

$$\Sigma(\mathbf{A}, \mathbf{b}) = \{\tilde{\mathbf{x}} \in \mathsf{R}^n \mid (\exists\, \tilde{\mathbf{A}} \in \mathbf{A})\,(\exists\, \tilde{\mathbf{b}} \in \mathbf{b})\; \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}\}. \tag{7}$$

It is sometimes called a *united solution set* and denoted by $\Sigma_{\exists\exists}(\mathbf{A}, \mathbf{b})$, as there are other possible definitions of solutions to Eq. (6), see e.g. [32, 33].

Usually the set $\Sigma(\mathbf{A}, \mathbf{b})$ is not an interval vector, and can be of quite complicated shape (in general, not necessarily convex, connected, or bounded). It is connected and bounded if the matrix $\mathbf{A}$ is regular. In this case, it constitutes an n-dimensional polyhedron which is a sum of at most 2^n convex polyhedrons obtained as intersections of the set $\Sigma(\mathbf{A}, \mathbf{b})$ with every of the 2^n orthants of the solution space $Ox_1 \ldots x_n$. The *convex hull* conv $\Sigma(\mathbf{A}, \mathbf{b})$ of this set is a minimal convex polyhedron containing $\Sigma(\mathbf{A}, \mathbf{b})$; as can be easily seen, the vertices of the convex hull constitute a subset of vertices of the solution set.

Another important and useful characterisation of the solution set, valid also for singular matrices $\mathbf{A}$, was given by Oettli and Prager [22] (another proof of the formula was later given by Rohn [26]):

$$\Sigma(\mathbf{A}, \mathbf{b}) = \{\tilde{\mathbf{x}} \in \mathsf{R}^n \mid \left|\check{\mathbf{A}}\tilde{\mathbf{x}} - \check{\mathbf{b}}\right| \le (\operatorname{rad}\mathbf{A})\left|\tilde{\mathbf{x}}\right| + \operatorname{rad}\mathbf{b}\}. \tag{8}$$

Calculating (and representing) the solution set $\Sigma(\mathbf{A}, \mathbf{b})$ may be quite hard and impractical, especially for larger n. Therefore, for many practical purposes we are satisfied with the *interval enclosure* of the set. The smallest (tightest) enclosure is the *hull* of the set:

$$\text{hull}\,\Sigma(\mathbf{A}, \mathbf{b}) = [\inf \Sigma(\mathbf{A}, \mathbf{b}), \sup \Sigma(\mathbf{A}, \mathbf{b})].$$

Obviously, hull $\Sigma(\mathbf{A}, \mathbf{b})$ = hull conv $\Sigma(\mathbf{A}, \mathbf{b})$. This enclosure is also hard to calculate in general case (see the next section), hence a number of more effective algorithms producing less exact enclosures has been devised. Some of them are described and used in the sequel to solve certain linear mechanical problems.

3.1. Finding the exact hull of the solution set

There are several methods for obtaining the hull which is the *exact* (called also *optimal*) interval enclosure of the solution set $\Sigma(\mathbf{A}, \mathbf{b})$. However, it was proven that finding the hull is an inherently exponential complexity problem, see Sec. 5.1.2 for details. Fortunately, this is the worst-case behaviour—for many practical problems some of the algorithms of this class exhibit much better performance [2, 26].

3.1.1. Enumerating combinations of endpoints of interval coefficients (CEIC)

This method of calculating the hull, called also a *combinatorial method* [25], is very simple and easy to implement, hence often used as a reference algorithm during experiments with implementation and application of other, more intricate algorithms.

The CEIC algorithm is based on the theorem by Hartfiel [9] which states (after reformulation into our notation) that for a regular interval matrix $\mathbf{A}$:

$$\text{conv}\,\Sigma(\mathbf{A}, \mathbf{b}) = \text{conv}\,\Sigma(\partial\mathbf{A}, \partial\mathbf{b}),$$

where:

$$\Sigma(\partial\mathbf{A}, \partial\mathbf{b}) = \{\tilde{\mathbf{x}} \in \mathrm{R}^n \mid (\exists\, \tilde{\mathbf{E}} \in \partial\mathbf{A})\,(\exists\, \tilde{\mathbf{e}} \in \partial\mathbf{b})\; \tilde{\mathbf{E}}\tilde{\mathbf{x}} = \tilde{\mathbf{e}}\}. \tag{9}$$

Since for every S holds hull conv S = hull S, then hull $\Sigma(\mathbf{A}, \mathbf{b})$ = hull $\Sigma(\partial\mathbf{A}, \partial\mathbf{b})$. That is, the method works by computing standard numerical solutions for all 2^t = card $\partial\mathbf{A}$ · card $\partial\mathbf{b}$ real systems of equations for all combinations endpoints of the interval elements of the matrix $\mathbf{A}$ and vector $\mathbf{b}$ (where $t = t_{\mathbf{A}} + t_{\mathbf{b}}$ is the number of thick interval coefficients in them), and returning the interval envelope of the resulting set of solutions.

Obviously, the algorithm is of exponential complexity: in the worst case, when all intervals in $\mathbf{A}$ and $\mathbf{b}$ are thick, $t = n^2+n$ and the algorithm must solve 2^{n^2+n} linear systems of n equations, hence its practical value is small. However, when interval elements constitute a small fraction of the coefficients of the system (e.g., for sparse matrices or limited uncertainty in system parameters) it may become more useful than other algorithms of this type described in the subsequent sections.

As $\Sigma(\partial\mathbf{A}, \partial\mathbf{b}) \subset \Sigma(\mathbf{A}, \mathbf{b})$ and $\Sigma(\partial\mathbf{A}, \partial\mathbf{b})$ contains all extremal points of the solution set $\Sigma(\mathbf{A}, \mathbf{b})$, a Monte-Carlo random sampling of the set $\Sigma(\partial\mathbf{A}, \partial\mathbf{b})$ has good chances to show qualitatively the overall shape of the solution set $\Sigma(\mathbf{A}, \mathbf{b})$.

3.1.2. The Rohn sign-accord algorithm (RSA)

One way of improving the CEIC algorithm is to find a method to filter out as many as possible of those elements of $\partial\mathbf{A}$ and $\partial\mathbf{b}$ which do not lead to solutions occupying the extremal points of $\Sigma(\mathbf{A}, \mathbf{b})$. Such a filtering scheme was indeed found by Rohn [20, 26].

Let $J = \{ \mathbf{j} \in \mathrm{R}^n \mid |\mathbf{j}| = (1, 1, ..., 1)^{\mathrm{T}} \}$ denotes a set of all n-component vectors with components equal to +1 or –1. Obviously, card $J = 2^n$. For any vector $\mathbf{v} = (v_1, v_2, ..., v_n)^{\mathrm{T}}$, let $\mathbf{D}_{\mathbf{v}}$ = diag $(v_1, v_2, ..., v_n)$ denotes a diagonal $n \times n$ matrix with components of $\mathbf{v}$ along the diagonal. Then, let us form the following matrices:

$$\begin{cases} \mathbf{A}_{\mathbf{rc}} = \breve{\mathbf{A}} - \mathbf{D}_{\mathbf{r}} (\mathrm{rad}\, \mathbf{A}) \mathbf{D}_{\mathbf{c}}, \\ \mathbf{b}_{\mathbf{r}} = \breve{\mathbf{b}} + \mathbf{D}_{\mathbf{r}}\, \mathrm{rad}\, \mathbf{b}. \end{cases}$$

where $\mathbf{r}, \mathbf{c} \in J$. Obviously, using the definition of midpoint and radius of an interval matrix:

$$(\mathbf{A}_{\mathbf{rc}})_{ij} = \begin{cases} \underline{a}_{ij} & \text{if } r_i c_j = 1, \\ \overline{a}_{ij} & \text{if } r_i c_j = -1, \end{cases}$$

$$(\mathbf{b}_{\mathbf{r}})_i = \begin{cases} \overline{b}_i & \text{if } r_i = 1, \\ \underline{b}_i & \text{if } r_i = -1, \end{cases}$$

hence $\mathbf{A}_{\mathbf{rc}}$ and $\mathbf{b}_{\mathbf{r}}$ are boundary matrices: $\mathbf{A}_{\mathbf{rc}}, \in \partial\mathbf{A}$, $\mathrm{b}_{\mathbf{r}} \in \partial\mathbf{b}$. Now putting:

$$\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}}) = \{ \tilde{\mathbf{x}} \in \mathrm{R}^n \mid (\exists \mathbf{r}, \mathbf{c} \in J)\ \mathbf{A}_{\mathbf{rc}} \tilde{\mathbf{x}} = \mathbf{b}_{\mathbf{r}} \}. \tag{10}$$

we see that $\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}}) \subseteq \Sigma(\partial\mathbf{A}, \partial\mathbf{b}) \subset \Sigma(\mathbf{A}, \mathbf{b})$. Moreover, as proven by Rohn [20, 26], we have again, for any regular matrix $\mathbf{A}$, conv $\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}})$ = conv $\Sigma(\mathbf{A}, \mathbf{b})$, hence also hull $\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}})$ = hull $\Sigma(\mathbf{A}, \mathbf{b})$. Since the number of different pairs of vectors $\mathbf{r}, \mathbf{c} \in J$ equals $2^n \cdot 2^n = 2^{2n}$, we have card $\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}}) \le 2^{2n}$ (some of the solutions for different pairs of vectors $\mathbf{r}, \mathbf{c}$ may be the same). However, since we do not know in advance which solutions will repeat in $\Sigma(\mathbf{A}_{\mathbf{rc}}, \mathbf{b}_{\mathbf{r}})$, finding this set, and hence the hull, would still require solving (exactly) 2^{2n} linear systems of n equations. That, though still exponential with the size of the problem, is already a huge improvement in comparison with the worst-case complexity of the CEIC algorithm, namely by a factor of $2^{n^2+n} / 2^{2n} = 2^{n^2-n}$.

However, still further improvement is possible. Namely, as Rohn showed in [26], each vertex of conv $\Sigma(\mathbf{A}, \mathbf{b})$ satisfies the equation $\mathbf{A}_{\mathbf{rc}} \tilde{\mathbf{x}} = \mathbf{b}_{\mathbf{r}}$ where $\mathbf{c} = \mathrm{sgn}\ \tilde{\mathbf{x}}$ (i.e., $\mathbf{D}_{\mathbf{c}} \tilde{\mathbf{x}} \ge 0$). Hence, if we could

somehow find for every vector $\mathbf{r} \in J$ such a vector $\mathbf{c} \in J$ that the solution $\tilde{\mathbf{x}}$ to the linear systems of equations $\mathbf{A_{rc}}\tilde{\mathbf{x}} = \mathbf{b_r}$ will have the property $\operatorname{sgn} \tilde{\mathbf{x}} = \mathbf{c}$ (for the purpose of this analysis we can safely assume that $\operatorname{sgn} 0 = 1$), then the total number of linear systems to solve would be only 2^n (i.e., the cardinality of J). This leads to the following algorithm:

Rohn Sign-Accord algorithm (RSA):

For every $\mathbf{r} \in J$ do:

Step 0: Select a $\mathbf{c} \in J$ (recommended: $\mathbf{c} = \operatorname{sgn}(\check{\mathbf{A}}^{-1}\mathbf{b_r})$).

Step 1: Solve $\mathbf{A_{rc}}\tilde{\mathbf{x}} = \mathbf{b_r}$.

Step 2: If $\operatorname{sgn} \tilde{\mathbf{x}} = \mathbf{c}$, register $\tilde{\mathbf{x}}$ and go to next $\mathbf{r}$, otherwise:

Step 3: Find $k = \min \{ j \mid \operatorname{sgn} \tilde{\mathbf{x}}_j \neq \mathbf{c}_j \}$.

Step 4: Set $\mathbf{c}_k = -\mathbf{c}_k$ and go to *Step 1*.

The algorithm finds all 2^n vertices of $\operatorname{conv} \sum(\mathbf{A}, \mathbf{b})$ in a finite number of steps regardless of the choice of initial vector $\mathbf{c} \in J$ in *Step 0*. However, the recommendation included there is quite important. As it was shown by Rohn [26], while the worst-case complexity of the algorithm is still of the order of 2^{2n} (with the inner loop [*Step 1 – Step 4*] traversed 2^n times for every $\mathbf{r}$), starting with the recommended value of $\mathbf{c}$ leads for many types of problems to much smaller complexity, often as low as 2^n (with the inner loop traversed only once for every $\mathbf{r}$). The hull of $\sum(\mathbf{A}, \mathbf{b})$ is then easily obtained as the hull of the 2^n solutions registered at *Step 2* of the algorithm. In fact, compiling the whole set of solutions is not necessary for calculation of the hull, as it can be made "on the fly", by simply updating current minimal and maximal values of vector $\tilde{\mathbf{x}}$ components with every new solution found at *Step 2*.

The algorithm can be also extended into a more complicated form capable of testing for regularity of the matrix $\mathbf{A}$ along the way [20, 26].

A problem with this algorithm may arise for systems in which components of some endpoint solution(s) are near to zero. Then possible roundoff errors may produce wrong value for $\operatorname{sgn} \tilde{\mathbf{x}}$ at *Step 2*, which may lead to infinite looping of the algorithm [20].

3.1.2. The linear programming method (LPM)

As the problem of finding the hull is in fact a problem of finding extremal values of some set of numbers, one may try to formulate it as an optimisation problem—minimisation (or maximisation) of appropriate objective function subject to appropriate constraints. The resulting linear programming method (LPM) has been first formulated in [21] and then used, among others, in [12, 23, 25].

The derivation of the method starts from Oettli and Prager [22] characterisation of the solution set $\sum(\mathbf{A}, \mathbf{b})$ (see Eq. (8)):

$$\tilde{\mathbf{x}} \in \sum(\mathbf{A}, \mathbf{b}) \Leftrightarrow \left| \check{\mathbf{A}}\tilde{\mathbf{x}} - \check{\mathbf{b}} \right| \leq (\operatorname{rad} \mathbf{A}) \left| \tilde{\mathbf{x}} \right| + \operatorname{rad} \mathbf{b} .$$

We may get rid of the first absolute value in the formula rewriting the inequality above as an equivalent system of two inequalities:

$$\begin{cases} \check{\mathbf{A}}\mathbf{x} - \check{\mathbf{b}} \leq (\operatorname{rad} \mathbf{A}) \left| \tilde{\mathbf{x}} \right| + \operatorname{rad} \mathbf{b}, \\ -\check{\mathbf{A}}\mathbf{x} + \check{\mathbf{b}} \leq (\operatorname{rad} \mathbf{A}) \left| \tilde{\mathbf{x}} \right| + \operatorname{rad} \mathbf{b}. \end{cases}$$

They still are non-linear with respect to $\tilde{\mathbf{x}}$. However, if we know the sign $\mathbf{s}$ of the solution $\tilde{\mathbf{x}}$, $\mathbf{s} = \operatorname{sgn} \tilde{\mathbf{x}}$, we can write $\tilde{\mathbf{x}} = \mathbf{D_s} |\tilde{\mathbf{x}}|$, where $\mathbf{D_s} = \operatorname{diag}(s_1, s_2, \ldots, s_n)$ (see previous Section; again we can safely assume that $\operatorname{sgn} 0 = 1$). Then:

$$\begin{cases} (\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} - \operatorname{rad}\mathbf{A})\,|\tilde{\mathbf{x}}| \le \breve{\mathbf{b}} + \operatorname{rad}\mathbf{b}, \\ -(\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} + \operatorname{rad}\mathbf{A})\,|\tilde{\mathbf{x}}| \le -\breve{\mathbf{b}} + \operatorname{rad}\mathbf{b}, \end{cases}$$

and finally:

$$\begin{cases} (\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} - \operatorname{rad}\mathbf{A})\,|\tilde{\mathbf{x}}| \le \overline{\mathbf{b}}, \\ (\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} + \operatorname{rad}\mathbf{A})\,|\tilde{\mathbf{x}}| \ge \underline{\mathbf{b}}. \end{cases}$$

The matrices on the left-hand side of the inequalities above are real matrices build from endpoints of the original interval matrix $\mathbf{A}$, but they are not boundary matrices, since some coefficients also change sign:

$$(\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} - \operatorname{rad}\mathbf{A})_{ij} = \begin{cases} \underline{a}_{ij} & \text{if } (\mathbf{D}_\mathbf{s})_{jj} = 1, \\ -\overline{a}_{ij} & \text{if } (\mathbf{D}_\mathbf{s})_{jj} = -1, \end{cases}$$

$$(\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} + \operatorname{rad}\mathbf{A})_{ij} = \begin{cases} \overline{a}_{ij} & \text{if } (\mathbf{D}_\mathbf{s})_{jj} = 1, \\ -\underline{a}_{ij} & \text{if } (\mathbf{D}_\mathbf{s})_{jj} = -1. \end{cases}$$

The above inequalities are now linear with respect to $|\in\tilde{\mathbf{x}}|$, hence putting $\tilde{\mathbf{x}}' = |\in\tilde{\mathbf{x}}|$, we can formulate the following two coupled *linear programming problems* allowing us to find minimal and maximal values of $\tilde{\mathbf{x}}$ in any given orthant of the solution space $Ox_1 \ldots x_n$ defined by the sign vector $\mathbf{s}$ [21]:

Finding upper bound $\tilde{\mathbf{x}}^+$:
Maximise $f(\tilde{\mathbf{x}}') = \tilde{\mathbf{x}}'$ subject to $3n$ linear constraints of Eq. (11),
then put $\tilde{\mathbf{x}}^+ = \mathbf{D}_\mathbf{s} \max \tilde{\mathbf{x}}'$.

Finding lower bound $\tilde{\mathbf{x}}^-$:
Minimise $f(\tilde{\mathbf{x}}') = \tilde{\mathbf{x}}'$ subject to $3n$ linear constraints of Eq. (11),
then put $\tilde{\mathbf{x}}^- = \mathbf{D}_\mathbf{s} \min \tilde{\mathbf{x}}'$.

$$\begin{cases} \tilde{\mathbf{x}}' \ge 0, \\ (\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} - \operatorname{rad}\mathbf{A})\tilde{\mathbf{x}}' \le \overline{\mathbf{b}}, \\ (\breve{\mathbf{A}}\mathbf{D}_\mathbf{s} + \operatorname{rad}\mathbf{A})\tilde{\mathbf{x}}' \ge \underline{\mathbf{b}}. \end{cases} \tag{11}$$

In the case the whole set of solutions $\Sigma(\mathbf{A}, \mathbf{b})$ resides in one orthant defined by the sign vector $\mathbf{s}$, the interval vector $\mathbf{h} = [\inf(\tilde{\mathbf{x}}^-, \tilde{\mathbf{x}}^+), \sup(\tilde{\mathbf{x}}^-, \tilde{\mathbf{x}}^+)]$ provides the required hull. Otherwise, all orthants containing solutions should be inspected, that is, the linear programming problems as defined above should be solved for each of these orthants (as selected with appropriate choice of vectors $\mathbf{s}$) with the hull of such obtained set of solutions giving the required result. Hence, the worst case complexity of the linear programming approach is also exponential—it may require solving as many as 2^n (the number of orthants) coupled linear programming problems. Total complexity depends also on the complexity of the linear programming algorithm, especially on the number of evaluations of the four matrix multiplication constraints in Eq. (11).

3.2. Interval estimates of the hull of the solution set

Trading computational complexity for estimation accuracy, it is possible to derive polynomial-time algorithms producing interval estimations that are larger than the hull of the solution set. There is quite a number of such algorithms, of various complexity, accuracy, applicability conditions and effectiveness, scattered over a wide selection of literature sources. Only a few were as yet tested by us and are described here. Applying them to real-world problems is hindered by several obstacles, among others unclear

formulation (from algorithmic and computational point of view) and lack of analysis of their overestimation properties and/or computational complexity (see Sec. 5.1. below for further discussion of this topic).

Besides the two methods described below, worth mentioning is another effective algorithm due to Rohn [27, 29], based on an earlier idea by Hansen.

3.2.1. Preconditioning

Trying to solve the original system of interval linear equations $\mathbf{Ax} = \mathbf{b}$ by some interval extension of the standard methods for solving systems of linear equations (e.g., through calculation of interval inverse matrix), we get usually large overestimation of the hull of the solution set $\Sigma(\mathbf{A}, \mathbf{b})$, mostly due to certain inconvenient features of interval arithmetic (see discussion in Sec. 5.2 below). A common method for decreasing the overestimation is to transform the set of equations to a more tractable form. This is usually called *preconditioning* and is based on the following considerations (see Neumaier [20] for appropriate supporting theorems).

First, for an interval matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$, if $\mathbf{CAC}'$ is an *H*-matrix for some real matrices $\mathbf{C}, \mathbf{C}' \in \mathrm{R}^{n\times n}$, then $\mathbf{A}$ is strongly regular and:

$$\begin{aligned} &\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \{\mathbf{C}'\tilde{\mathbf{x}} \in \mathrm{R}^n \mid \tilde{\mathbf{x}} \in \Sigma(\mathbf{CAC}', \mathbf{Cb})\}, \\ &\operatorname{hull} \Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{C}' (\operatorname{hull} \Sigma(\mathbf{CAC}', \mathbf{Cb})). \end{aligned} \tag{12}$$

That is, both the solution set and hull of the original systems of equations are contained in the solution set and the hull, respectively, of another system of equations (the *preconditioned* system), namely the system $\mathbf{CAC}'\mathbf{x} = \mathbf{Cb}$, only transformed by a linear transformation given by $\mathbf{C}'$. Moreover:

$$\left|\operatorname{hull} \Sigma(\mathbf{A}, \mathbf{b})\right| \le \left|\mathbf{C}'\right| \langle \mathbf{CAC}' \rangle^{-1} \left|\mathbf{Cb}\right|. \tag{13}$$

Second, as $\mathbf{CAC}'$ is required to be an *H*-matrix, then also $\langle \mathbf{CAC}' \rangle \mathbf{u} > 0$ for some $\mathbf{u} > 0$ (see the definition of the *H*-matrix in Sec. 2.2) and

$$\operatorname{hull} \Sigma(\mathbf{A}, \mathbf{b}) \subseteq \left\|\mathbf{Cb}\right\|_{\mathbf{v}} \cdot \left|\mathbf{C}'\right| [-\mathbf{u}, \mathbf{u}], \tag{14}$$

where $\mathbf{v}$ must satisfy $\langle \mathbf{CAC}' \rangle \mathbf{u} \ge \mathbf{v} > 0$.

As for the preconditioning matrices $\mathbf{C}$ and $\mathbf{C}'$, usually the choices $\mathbf{C} = \breve{\mathbf{A}}^{-1}$ and $\mathbf{C}' = \mathbf{I}$ are made. Then the condition of $\mathbf{CAC}' = \breve{\mathbf{A}}^{-1}\mathbf{A}$ being an *H*-matrix is fulfilled whenever $\mathbf{A}$ is strongly regular (i.e., $\breve{\mathbf{A}}^{-1}\mathbf{A}$ is regular). They are optimal in the sense of Eq. (13), where they give the minimal upper bound in a general case. For this choice, Eqs. (12, 14) obtain simpler formulations:

$$\begin{aligned} &\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \Sigma(\breve{\mathbf{A}}^{-1}\mathbf{A}, \breve{\mathbf{A}}^{-1}\mathbf{b}), \\ &\operatorname{hull} \Sigma(\mathbf{A}, \mathbf{b}) \subseteq \operatorname{hull} \Sigma(\breve{\mathbf{A}}^{-1}\mathbf{A}, \breve{\mathbf{A}}^{-1}\mathbf{b})). \end{aligned} \tag{12′}$$

and:

$$\operatorname{hull} \Sigma(\mathbf{A}, \mathbf{b}) \subseteq \left\| \breve{\mathbf{A}}^{-1}\mathbf{b} \right\|_{\mathbf{v}} \cdot [-\mathbf{u}, \mathbf{u}]. \tag{14′}$$

where $\mathbf{u}$ and $\mathbf{v}$ must satisfy $\mathbf{u} > 0$ and $\langle \breve{\mathbf{A}}^{-1}\mathbf{A} \rangle \mathbf{u} \ge \mathbf{v} > 0$.

This preconditioning is called *preconditioning with the midpoint inverse* and constitutes the first step of many algorithms for estimation of the hull of $\Sigma(\mathbf{A}, \mathbf{b})$. However, as it only optimises the upper bound in Eq. (13), in some special cases other preconditioning matrices may give better estimation of the solution set and its hull, according to Eq. (12).

The estimate given by Eq. (14′) is rather crude (one indication of that is that it must be symmetrical around zero, which leads to great overestimations when $\Sigma(\mathbf{A}, \mathbf{b})$ is located unsymmetrically and far from

the origin). It is also linearly dependent on the norm of the vector **b**. Hence, one of the possibilities of sharpening the estimate is to transform the system into another one with smaller values in the right-hand side vector. As can be shown, if we find some particular solution $\tilde{\mathbf{x}} \in \Sigma(\mathbf{A}, \mathbf{b})$, we have hull $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \tilde{\mathbf{x}}$ + hull $\Sigma(\mathbf{A}, \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}})$. If $\tilde{\mathbf{x}}$ is an appropriately chosen solution (a good choice is the solution of the midpoint system: $\tilde{\mathbf{x}} = \check{\mathbf{A}}^{-1}\check{\mathbf{b}}$), the residual correction vector $\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$ has a chance to be much closer to zero than **b**. Hence, the final estimation:

$$\text{hull } \Sigma(\mathbf{A}, \mathbf{b}) \subseteq \tilde{\mathbf{x}} + \left\| \check{\mathbf{A}}^{-1}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}) \right\|_{\mathbf{v}} \cdot [-\mathbf{u}, \mathbf{u}] \qquad (14'')$$

may be much sharper (note that it is no longer necessarily symmetric around zero, instead being symmetric around $\tilde{\mathbf{x}}$).

To get still better results, the estimate of Eq. (14″) is usually used only as a starting point to be improved by some iterative method, like Gauss-Seidel iteration, see the next Section.

3.2.2. Preconditioned Gauss-Seidel Iteration (PGSI)

Let us start from writing the system $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ explicitly as a set of equations:

$$\sum_{k=1}^{n} \tilde{a}_{ik}\tilde{x}_k = \tilde{b}_i, \text{ for } i = 1, ..., n.$$

Assuming that $\tilde{a}_{ii} \neq 0$, we can solve the i-th equation for the i-th variable. This gives:

$$\tilde{x}_i = (\tilde{b}_i - \sum_{k \neq i} \tilde{a}_{ik}\tilde{x}_k) / \tilde{a}_{ii}.$$

Hence, provided the initial interval enclosure vector **x** containing $\tilde{\mathbf{x}}$ is known and $0 \notin a_{ii}$:

$$\tilde{x}_i \in x_i' = (b_i - \sum_{k \neq i} a_{ik}x_k) / a_{ii},$$

so that in this way we get the i-th component of the new, possibly improved enclosure vector **x′**. As we can repeat this procedure for all $i = 1, ..., n$, the full new enclosure vector **x′** can be obtained. Since this works for all $\tilde{\mathbf{x}} \in \mathbf{x}$ with $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, $\tilde{\mathbf{A}} \in \mathbf{A}$, and $\tilde{\mathbf{b}} \in \mathbf{b}$, we have:

$$\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x} \subseteq \mathbf{x}' \cap \mathbf{x},$$

i.e. we can obtain a new enclosure which is at least not worse than the previous one. Moreover, this means the method works also when $\Sigma(\mathbf{A}, \mathbf{b})$ is unbounded (i.e., when **A** is singular), provided we are interested in finding a better enclosure for a *part* of the solution set included in some initial box **x**.

We can improve the enclosure faster by using on the i-th step the already obtained new enclosures from the previous steps. This finally leads to the basic iteration step in the form:

$$y_i := ((b_i - \sum_{k<i} a_{ik}\, y_k - \sum_{k>i} a_{ik}\, x_k) / a_{ii}) \cap x_i, \text{ for } i = 1, ..., n.$$

The above works well when $0 \notin a_{ii}$ for all i. Fortunately, it can be made to work (still with possible reduction of the width of the component x_i) also for diagonal coefficients containing zero, using the simple technique described in [8] and based on Kahan arithmetic rules.

Finally, integrating preconditioning with the above iterative improvement method we may formulate:

Preconditioned Gauss-Seidel Iteration algorithm (PGSI):

Step 1: Calculate the midpoint inverse matrix $\breve{\mathbf{A}}^{-1}$.

Step 2: Find a solution $\tilde{\mathbf{x}} = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}$ to the midpoint system of equations $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$.

Step 3: Replace the initial system with the new system $\mathbf{Ax} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$.

Step 4: Precondition the new system with the midpoint inverse matrix, obtaining: $\breve{\mathbf{A}}^{-1}\mathbf{Ax} = \breve{\mathbf{A}}^{-1}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}})$.

Step 5: Find the vectors **u** and **v** necessary to estimate the initial enclosure of the solution set (see [20] for justification of this particular choice):

$$\mathbf{u} = \langle \breve{\mathbf{A}}^{-1}\mathbf{A}\rangle^{-1}(1, 1, ..., 1)^{\mathrm{T}}, \quad \mathbf{v} = \langle \breve{\mathbf{A}}^{-1}\mathbf{A}\rangle\,\mathbf{u}.$$

Step 6: Calculate the initial enclosure $\mathbf{x}^{(0)}$ of the solution set:

$$\mathbf{x}^{(0)} = \tilde{\mathbf{x}} + \left\| \breve{\mathbf{A}}^{-1}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}) \right\|_{\mathbf{v}} \cdot [-\mathbf{u}, \mathbf{u}].$$

Step 7: Improve the enclosure by repeated application of the Gauss-Seidel iteration step, calculating the vectors $\mathbf{x}^{(l)}$, $l = 0, 1, 2, ...$, by:

$$\mathbf{x}^{(l+1)} := ((\mathrm{b}_i - \sum_{k<i} \mathrm{a}_{ik}\, x_k^{(l+1)} - \sum_{k>i} \mathrm{a}_{ik}\, x_k^{(l)}) / \mathrm{a}_{ii}) \cap x_i^{(l)},$$

until some suitable stopping criterion is met (e.g., until the enclosure $\mathbf{x}^{(l)}$ ceases to improve).

The above algorithm works for strongly regular matrices **A**; it gives especially good results when **A** is an *H*-matrix. If **A** is not an *H*-matrix, it may happen than some arbitrarily large initial estimation **x** is not improved at all by the Gauss-Seidel iteration step [20].

3.2.3. Preconditioned Gauss Elimination (PGE)

This algorithm is an interval version of the classic Gauss elimination algorithm known from standard numerical analysis [20, 24]. Let us briefly review the derivation of the standard method before going into its interval version.

Let $\mathbf{A} \in \mathrm{R}^{n\times n}$ and $\mathbf{b} \in \mathrm{R}^{n}$. Starting with the system:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

we subtract a suitable multiple of the first row from the other rows such that the subdiagonal entries of the first column become zero. This requires that $a_{11} \neq 0$; then the multiplication factor for the i-th row is $l_{i1} = a_{i1} / a_{11}$, with $i > 1$. After the subtraction we have the reduced system:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{bmatrix}, \tag{15}$$

where:

$$a_{ik}^{(1)} = a_{ik} - l_{ik}a_{1k} = a_{ik} - a_{i1}a_{11}^{-1}a_{1k},$$
$$b_i^{(1)} = b_i - l_{i1}b_1 = b_i - a_{i1}a_{11}^{-1}b_1,$$

for $i, k = 2, ..., n$. After the other unknowns, i.e. $x^{(1)} = (x_2, ..., x_n)^{\mathrm{T}}$ are determined from the smaller system $\mathbf{A}^{(1)}\mathbf{x}^{(1)} = \mathbf{b}^{(1)}$, the first variable is obtained from the first equation as:

$$x_1 = (b_1 - \sum_{k>1} a_{1k}\, x_k) \,/\, a_{11}\,.$$

As long as the corresponding diagonal elements $a_{jj}^{(j-1)}$ remain nonzero, we may eliminate further variables in the same way by:

$$\begin{cases} l_{ij} = a_{ij}^{(j-1)} \big/ a_{jj}^{(j-1)}, \\ a_{ik}^{(j)} = a_{ik}^{(j-1)} - l_{ij}a_{jk}^{(j-1)}, \\ b_i^{(j)} = b_i^{(j-1)} - l_{ij}b_j^{(j-1)}. \end{cases}$$

The variable x_j can then be obtained from $x_{j+1}, ..., x_n$ by:

$$x_j = (\, b_j^{(j-1)} - \sum_{k>j} a_{jk}^{(j-1)}\, x_k) \,/\, a_{jj}^{(j-1)}, \text{ for } j = 1, 2, ..., n.$$

The first step is subsumed by the above after putting $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{b}^{(0)} = \mathbf{b}$.

The multiplication factors l_{ij}, $i > j$, and coefficients $u_{jk} = a_{jk}^{(j-1)}$, $k \geq j$, occurring in the above formulas can be combined into triangular matrices:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{21} & 1 & \cdots & 0 & 0 \\ & l_{32} & \ddots & \vdots & \vdots \\ \vdots & \vdots & & 1 & 0 \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix}, \qquad \mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & & u_{1n} \\ 0 & u_{22} & \cdots & & u_{2n} \\ 0 & 0 & \ddots & & \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & u_{nn} \end{bmatrix}.$$

such that $\mathbf{A} = \mathbf{LU}$, $\mathbf{Ly} = \mathbf{b}$, and $\mathbf{Ux} = \mathbf{y}$. Thus, Gauss elimination consists in a factorisation of the matrix $\mathbf{A}$ into the product of two triangular matrices $\mathbf{L}$ and $\mathbf{U}$ (the so-called *LU*-decomposition [24]), reducing the solution of the original system of equations to two triangular systems $\mathbf{Ly} = \mathbf{b}$ and $\mathbf{Ux} = \mathbf{y}$ which can be easily solved by *forward substitution* and *backward substitution* as follows:

$$\begin{aligned} y_i &= b_i - \sum_{j<i} l_{ij} y_j && \text{for } i = 1,..., n, \\ x_i &= (y_i - \sum_{k>i} u_{ik} x_k) \,/\, u_{ii} && \text{for } i = n, n-1, ..., 1. \end{aligned} \tag{16}$$

The coefficients of $\mathbf{L}$ and $\mathbf{U}$ can be calculated from:

$$\begin{aligned} l_{ik} &= (a_{ik} - \sum_{j>k} l_{ij}\, u_{jk}) \,/\, u_{kk} && \text{for } i > k, \\ u_{ik} &= a_{ik} - \sum_{j<i} l_{ij}\, u_{jk} && \text{for } i \leq k. \end{aligned} \tag{17}$$

If $\mathbf{A}$ is regular, then even when $a_{jj}^{(j-1)} = 0$ for some j, the process can be finished successfully after a suitable permutation of rows or columns of $\mathbf{A}^{(j)}$.

For interval case, when $\mathbf{A} \in \mathbb{IR}^{n\times n}$ and $\mathbf{b} \in \mathbb{IR}^{n}$, the algorithm can proceed in much the same way (according to Eq. (17)), provided all $a_{jj}^{(j-1)} = u_{jj}$ do not contain zero. However, due to the properties of interval arithmetic, in general you cannot always zero the coefficients under the diagonal in Eq. (15). Hence, the product of triangular matrices $\mathbf{L}$ and $\mathbf{U}$ defined according to Eq. (17) does not equal $\mathbf{A}$; instead, we have only an inclusion $\mathbf{A} \subseteq \mathbf{LU}$. Similarly, we have only inclusions $\mathbf{Ly} \supseteq \mathbf{b}$ and $\mathbf{Ux} \supseteq \mathbf{y}$ instead of equalities. Hence, the resulting interval vector $\mathbf{x}$ will be in general wider than the hull of the solution set. Moreover, when some u_{jj} contains zero, permutation of the matrix does not always help, even when $\mathbf{A}$ is regular [20]. The algorithm always terminates, however, when $\mathbf{A}$ is an *H*-matrix. When $\mathbf{A}$ is an *M*-matrix and $\mathbf{b} \geq 0$ it produces the exact hull of the solution set. Again, since when $\mathbf{A}$ is strongly regular the preconditioned matrix $\breve{\mathbf{A}}^{-1}\mathbf{A}$ is an *H*-matrix, it gives good results when combined with preconditioning, often better than Gauss-Seidel iteration. In particular it has the quadratic approximation property, that is, the overestimation is of order $\mathrm{O}(\varepsilon^2)$ with the size of $\mathbf{b}$, which is advantageous when $|\mathbf{b}| < 1$.

The PGE algorithm can be then formulated as follows:

Preconditioned Gauss Elimination algorithm (PGE):

Step 1: Calculate the midpoint inverse matrix $\breve{\mathbf{A}}^{-1}$.

Step 2: Precondition the system with the midpoint inverse matrix, obtaining: $\breve{\mathbf{A}}^{-1}\mathbf{Ax} = \breve{\mathbf{A}}^{-1}\mathbf{b}$.

Step 3: Perform the *LU*-decomposition of the preconditioned matrix, so that: $\breve{\mathbf{A}}^{-1}\mathbf{A} \subseteq \mathbf{LU}$, see Eq. (17).

Step 4: Calculate the vector $\mathbf{y}$ (such that $\mathbf{Ly} = \breve{\mathbf{A}}^{-1}\mathbf{b}$) by forward substitution.

Step 5: Calculate the enclosure of the solution set, namely the vector $\mathbf{x}$ (such that $\mathbf{Ux} = \mathbf{y}$) by backward substitution.

4. MECHANICAL EXAMPLES

Our interest with interval methods turned up mainly as a result of our research on hybrid systems for qualitative and quantitative analysis of mechanical systems [11]. The hybrid expert system for analysis of truss structures we are currently implementing [16, 17] contains thus an interval-based module providing a qualitative-numerical analysis of the trusses. The truss structure examples given below were analysed using that system. The frame example is a simple textbook example [4] used in the experiments with fuzzy set approach to mechanical structures [23, 34]; we are considering to extend our hybrid system to incorporate analysis of frame structures in the future. For simplicity, the examples are restricted to planar structures. An extension to spatial structures is quite straightforward.

4.1. Truss structures

A *truss structure* is a mechanical system build from elastic elongated *bars* joined at *nodes* using flexible, rotary joints, and loaded by some external force(s) applied at its nodes. Some of the nodes can be supported, by *full support* (giving no degrees of freedom to the supported node), or *partial (sliding) support* (allowing the node to move along a specified line or within a specified plane). In such a structure there is no bending of bars, hence the bars carry only axial forces. A planar truss has all its bars, as well as the loads, placed in a single plane. An example truss is shown in Fig. 1a, with one external load and two supports (one full, one sliding).

4.1.1. The system of linear equations for the truss

The standard displacement method of truss analysis leads to a set of linear equations with node displacements in the bars as unknowns. The construction of this system of equations starts from two sets of equations: one relating magnitudes of axial forces P_{ij} in the bars to the displacements of the nodes, and the other coming from force equilibrium condition at every node:

$$\begin{aligned} &P_{ij} = s_{ij}(\Delta_{ij} + \Delta_{ji}) = s_{ij}((d_j^x - d_i^x)\cos\alpha_{ij} + (d_j^y - d_i^y)\sin\alpha_{ij}), \\ &\vec{F}_i + \sum_{j=1}^{n} \vec{P}_{ij} = 0, \end{aligned} \tag{18}$$

where indices $i, j = 1, ..., n$ denote the node numbers, $\vec{F}_i$ is the external load at the i-th node, $\vec{P}_{ij}$ is the reaction force exerted by the bar (ij) at the node i, $s_{ij} = A_{ij} E_{ij} / l_{ij}$ is the stiffness of the bar (ij), with A_{ij} being cross-sectional area, E_{ij} Young's modulus, and l_{ij} length of the bar (we set $s_{ij} = 0$ if there is no bar linking nodes i and j), while Δ_{ij} and Δ_{ji} are projections of the displacement vectors of node i and j on the bar (ij), see Fig. 1b. The above assumes small deformations and linear elastic material law, so that the elongation of the bar can be given by the simple formula (the first row of Eq. (18)) and change of

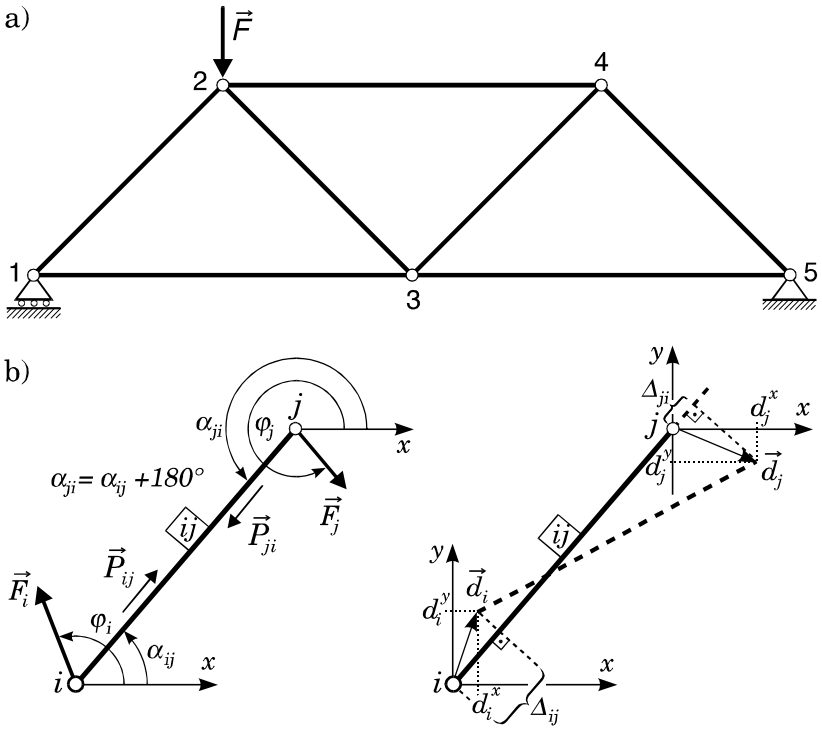


Fig. 1. An example of planar truss structure (a)
and a deformed bar between two nodes (b);
the deformation of the bar is greatly exaggerated in the drawing.

directions of the bars due to displacements of their ends can be neglected. Note also that $P_{ij} = |\vec{P}_{ij}|$, and by convention $P_{ij} > 0$ when the bar is stretched, and $P_{ij} < 0$ when the bar is compressed.

From Eq. (18) one easily obtains a set of $2n$ linear equations with displacement components d_i^x, d_i^y as unknowns, the right-hand side vector containing components of the external load vectors F_i^x, F_i^y, and

coefficients of the matrix of the system (the so-called *stiffness matrix* in this formulation) relating node displacements and loads. Omitting the derivation details that may be probably quite boring to most of the readers of this journal, we only show the final form of the linear system of equations **Ad** = **b** for the planar truss structure:

$$\begin{array}{c} \\ i^x \\ i^y \\ \\ j^x \\ j^y \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & i^x & i^y & & j^x & j^y & \end{array} \\ \begin{bmatrix} \ddots & \vdots & \vdots & & \vdots & \vdots & \\ \cdots & a_{ii}^{xx} & a_{ii}^{xy} & \cdots & a_{ij}^{xx} & a_{ij}^{xy} & \cdots \\ \cdots & a_{ii}^{yx} & a_{ii}^{yy} & \cdots & a_{ij}^{yx} & a_{ij}^{yy} & \cdots \\ & \vdots & \vdots & \ddots & \vdots & \vdots & \\ \cdots & a_{ji}^{xx} & a_{ji}^{xy} & \cdots & a_{jj}^{xx} & a_{jj}^{xy} & \cdots \\ \cdots & a_{ji}^{yx} & a_{ji}^{yy} & \cdots & a_{jj}^{yx} & a_{jj}^{yy} & \cdots \\ & \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \end{array} \times \begin{bmatrix} \vdots \\ d_i^x \\ d_i^y \\ \vdots \\ d_j^x \\ d_j^y \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ F_i \cos\varphi_i \\ F_i \sin\varphi_i \\ \vdots \\ F_j \cos\varphi_j \\ F_j \sin\varphi_j \\ \vdots \end{bmatrix} \begin{array}{l} \\ = b_i^x \\ = b_i^y \\ \\ = b_j^x \\ = b_j^y \\ \\ \end{array} \tag{19}$$

where:

$$\begin{array}{ll} a_{ii}^{xx} = \sum_j s_{ij} \cos^2 \alpha_{ij}, & a_{ij}^{xx} = -s_{ij} \cos^2 \alpha_{ij}, \\ a_{ii}^{xy} = a_{ii}^{xy} = \sum_j s_{ij} \sin \alpha_{ij} \cos \alpha_{ij}, & a_{ij}^{xy} = a_{ij}^{yx} = -s_{ij} \sin \alpha_{ij} \cos \alpha_{ij}, \\ a_{ii}^{yy} = \sum_j s_{ij} \sin^2 \alpha_{ij}, & a_{ij}^{yy} = -s_{ij} \sin^2 \alpha_{ij}, \\ \hline a_{ji}^{xx} = a_{ij}^{xx}, & a_{jj}^{xx} = \sum_i s_{ji} \cos^2 \alpha_{ji}, \\ a_{ji}^{xy} = a_{ji}^{yx} = a_{ij}^{xy} = a_{ij}^{yx}, & a_{jj}^{xy} = a_{jj}^{yx} = \sum_i s_{ji} \sin \alpha_{ji} \cos \alpha_{ji}, \\ a_{ji}^{yy} = a_{ij}^{yy}, & a_{jj}^{yy} = \sum_i s_{ji} \sin^2 \alpha_{ji}. \end{array} \tag{20}$$

Additional treatment is needed to account for the supported nodes, whose displacements are restricted partially or completely. For brevity, it will not be explained here in detail—it suffices to say that in the formulation used here it amounts to deleting some rows and corresponding columns from the matrices of the system of Eq. (19). After solving the system of Eqs. (19, 20) for node displacements, the axial forces in the bars can be calculated from the obtained displacements using the formulas from the first row of Eq. (18).

Note that in this formulation the matrix **A** of the system is symmetric and for every parameter of a given bar, more than one coefficient of the matrix depends on its value (more precisely, as many as 16 coefficients may depend on it). The significance of these observations will be made more clear in Sec. 5.2 below.

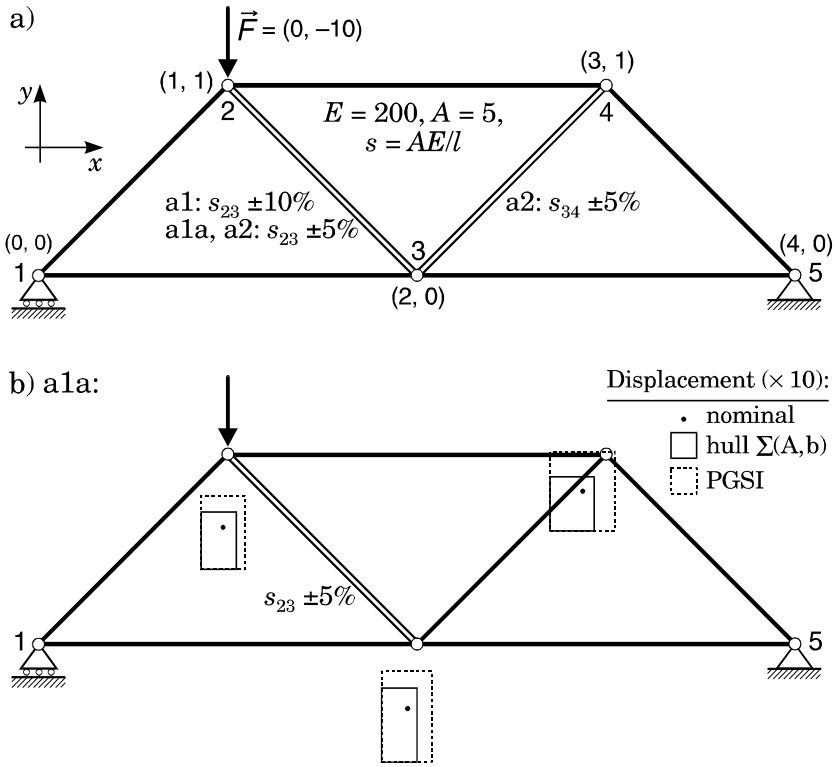


Fig. 2. Three cases of parameter uncertainties in the example truss (a)
and node displacement results (exaggerated ten times) for the case *a1a* (b).

4.1.2. Solving trusses with interval methods: an example

In this Section we discuss briefly results of interval analysis of the example truss, with three cases of parameter uncertainties, as depicted in Fig. 2a. We assume the stiffness of some of the bars to be uncertain, varying around the nominal value by ±5% (in cases *a1a* and *a2*) or by ±10% (case *a1*), for a single bar (in cases *a1* and *a1a*) or for two bars (case *a2*). Nominal values of parameters are also given.

For simplicity, the parameters of the truss are given here as dimensionless numbers, since the physical values are not relevant to our purely numerical analysis. The given values were chosen so as to be physically realistic when endowed with appropriate units.

Table 1. Interval estimates for the example truss (with 7×7 matrix);
the results with sign different than for the nominal solution are underlined.

a) *a1a*: s_{23} uncertain by ±5% [16 intervals and 14 zeros in the matrix]

	$\mathbf{d}_0$ [$\times 10^{-3}$]	hull $\sum$(A, b) [$\times 10^{-3}$]	wid$[*]/\mathbf{d}_0$ [%]	PGSI [$\times 10^{-3}$]	wid$[*]/\mathbf{d}_0$ [%]
d_1^x	–20	[–42.10, –12.03]	150	[–42.10, 2.10]	221
d_2^x	–2.5	[–14.12, 4.47]	744	[–14.12, 8.72]	914
d_2^y	–38.71	[–60.91, –30.71]	78	[–60.91, –22.03]	100
d_3^x	–5.0	[–18.81, –0.019]	376	[–18.81, 7.85]	533
d_3^y	–34.14	[–62.44, –23.26]	115	[–62.44, –14.52]	141
d_4^x	–12.5	[–29.65, –6.32]	187	[–29.65, 4.65]	274
d_4^y	–19.57	[–40.63, –11.98]	146	[–40.63, 1.49]	215

b) *a1*: s_{23} uncertain by ±10% [16 intervals and 14 zeros in the matrix]

	$\mathbf{d}_0$ [$\times 10^{-3}$]	hull $\sum$(A, b) [$\times 10^{-3}$]	wid $[*]/\mathbf{d}_0$ [%]	PGSI [$\times 10^{-3}$]	wid $[*]/\mathbf{d}_0$ [%]
d_1^x	–20	[–824.6, –6.00]	4090	[–824.6, 784.6]	8050
d_2^x	–2.5	[–425.6, 28.57]	18200	[–425.6, 419.7]	33800
d_2^y	–38.71	[–846.8, –24.65]	2120	[–846.8, 738.7]	4100
d_3^x	–5.0	[–507.8, 11.76]	10400	[–507.8, 495.7]	20100
d_3^y	–34.14	[–1064, –16.22]	3070	[–1064, 957.2]	5930
d_4^x	–12.5	[–636.7, –1.64]	5080	[–636.7, 611.7]	9990
d_4^y	–19.57	[–786.0, –6.24]	3980	[–786.0, 746.9]	7820

c) *a2*: s_{23} and s_{34} uncertain by ±5% [28 intervals and 12 zeros in the matrix]:
the matrix is singular (solution set unbounded).

The main results (node displacements) are summarised in Table 1. Figure 2b shows graphically the results (except for the node 1) for the case *a1a*; note that the displacements are exaggerated ten times for readability. The table compares the exact estimate of the solution set, i.e., the interval hull obtained by one of the algorithms of Sec. 3.1 with its estimate obtained by one of the algorithms of Sec. 3.2. The table contains also the *nominal solution* $\mathbf{d}_0$, i.e., the solution of the non-interval system of Eq. (19) for nominal values of parameters, as well as widths of the interval estimates relative to the nominal solution.

For the polynomial-complexity estimation methods (see Sec. 3.2), only the results of the PGSI algorithm are included in the table. The results of the PGE algorithm are practically the same, with only slight differences in a few places.

As can be seen from the tables, the resulting intervals are quite wide, compared to the width of the parameter intervals (the latter being equal to 10% for the cases *a1a* and *a2* and 20% for the case *a1*). For the case *a1a* (a single parameter uncertain by ±5%), the tightest estimate possible in this formulation (i.e., the hull) produces the relative width between 8 and as much as 74 times greater than the width of the parameter interval. The more effective polynomial-complexity methods add significant overestimation (with resulting widths up to about two times larger than for the hull). The widths for the case *a1* (with uncertainty increased to ±10%) are greater by more than an order of magnitude. It indicates that the width of the hull is very sensitive to an increase of the width of interval coefficients in the matrix. More annoying, notwithstanding the comparatively large width of the resulting intervals, in many cases even the sign (i.e., direction) of them becomes uncertain (indicated by underlining of the offending values)—the indeterminacy surely not justified by the uncertainty of the parameters. Hence, even the gross qualitative feature of the result has been estimated incorrectly in some places. For all other example trusses we have analysed, the general properties of the results are much the same. A systematic analysis of the dependence of widths of the results and overestimation effects on various types and extents of parameter uncertainties will be a subject of separate publication.

With two uncertain parameters (the case *a2* with ±5% uncertainty), the resulting interval matrix becomes singular, hence the set of solutions becomes unbounded (see Sec. 5.3 for more detailed discussion of this problem). As a result, the standard interval analysis does not produce any useful information in this case, despite the fact that real uncertainty of the resulting displacements (as it will be shown below in Sec. 5.2.3.1) remains quite small in this case too.

The results for this example show that standard interval analysis of mechanical structures with parameter uncertainties sometimes may not produce any useful results. The causes of the problem, together with the methods of reducing it, are discussed in more detail in Section 5 below.

4.2. Frames

A frame is a mechanical system more general than a truss structure. It is build from elastic elongated *beams* joined at *nodes* using both stiff joints (not allowed in truss structures) and possibly also rotary joints (like in truss structures), and loaded by some external forces applied at its nodes or distributed along the beams. Some of the nodes can be supported by *full support* (giving no degrees of freedom to the supported node), or *partial (sliding) support* (allowing the node to move along a specified line or within a specified plane). In frames a possibility of bending of beams is taken into account, hence the beams can carry also bending moments. A planar frame has all its beams, as well as the loads, placed in a single plane. An example planar frame is shown in Fig. 3a, with three types of support, and an external load distributed uniformly along the beam (24).

4.2.1. A frame example

For brevity, we shall not delve into general methods of analysing frame structures, restricting ourselves to showing the application of interval methods on the example frame (Fig. 3), after [23] (with several modifications), as it will suffice for our purposes here.

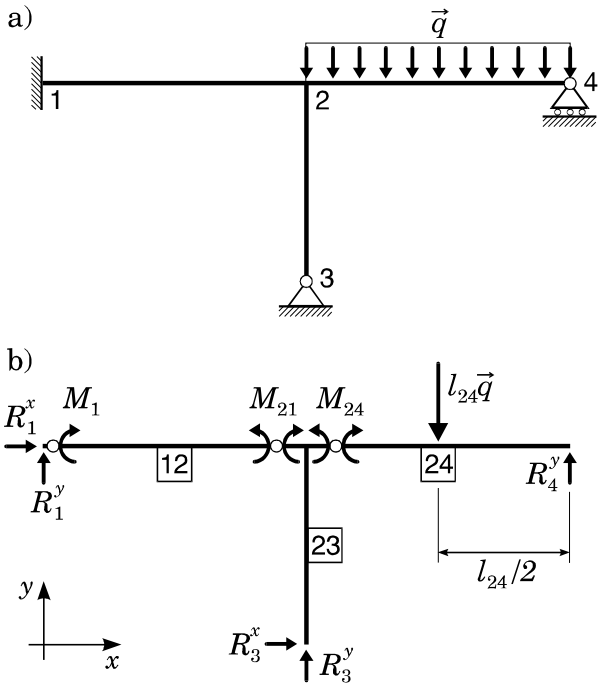


Fig. 3. An example of a frame (a)
and its fundamental system of internal parameters (b).

Assuming again small displacements and linear elastic material law and using the method of forces, the frame in question can be described by the set of equations which start from five equilibrium equations for forces and bending moments, see Fig. 3b:

$$\begin{cases} R_1^x + R_3^x = 0, \\ R_1^y + R_3^y + R_4^y - q l_{24} = 0, \\ -M_1 + R_4^y (l_{12} + l_{24}) + R_3^y l_{12} + R_3^x l_{23} - q l_{24} (l_{12} + \frac{1}{2} l_{24}) = 0, \\ -R_1^y l_{12} - M_1 + M_{21} = 0, \\ R_4^y l_{24} - \frac{1}{2} q l_{24}^2 - M_{24} = 0. \end{cases}$$

Then, the three canonical equations linking bending moments with material properties of the beams are given below, in matrix form (the derivation of the formulas is omitted here for brevity, see [4]). The beam properties are Young modulus E and momentum of inertia J of the beam cross-section.

$$\begin{bmatrix} \frac{l_{12}}{3E_{12}J_{12}} & \frac{l_{12}}{6E_{12}J_{12}} & 0 \\ \frac{l_{12}}{6E_{12}J_{12}} & \frac{l_{12}}{3E_{12}J_{12}} + \frac{l_{23}}{3E_{23}J_{23}} & \frac{-l_{23}}{3E_{23}J_{23}} \\ 0 & \frac{-l_{23}}{3E_{23}J_{23}} & \frac{l_{24}}{3E_{12}J_{12}} + \frac{l_{23}}{3E_{23}J_{23}} \end{bmatrix} \times \begin{bmatrix} M_1 \\ M_{21} \\ M_{24} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{-q l_{24}^3}{24 E_{24} J_{24}} \end{bmatrix}.$$

We can easily combine the above systems of equations into a single system of linear equations for reaction forces and bending moments. In this case the final matrix of the system is *not* symmetric, but similarly as in the case of trusses (Sec. 4.1.1), more than one coefficient of the matrix depends on the value of any given parameter. Moreover, also the elements of the right-hand side vector depend on parameters of the beams, not only on external loads (this is partly due to the presence of distributed load along one of the beams). Again, the significance of these observations will be made more clear in Sec. 5.2 below.

4.2.2. Interval solutions of the frame

To solve our example frame, we must fix some particular numerical values to its parameters, some of them possibly uncertain, hence modelled by intervals. Similarly as for truss structures, the parameters of the frame are given here as dimensionless numbers—it is assumed the given values are physically realistic when endowed with appropriate units. First, we assume here that all the beams have the same Young modulus E, but momentum of inertia J of beam cross-sections are related by the formula $J_{12} = J_{23} = 1.5 J_{24}$. Substituting that to the combined equations for the frame and making appropriate simplifications, we get the system:

$$\begin{bmatrix} 2l_{12} & l_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{12} & 2l_{12}+2l_{23} & -2l_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & -2l_{23} & 3l_{24}+2l_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & l_{12} & l_{12}+l_{24} & 0 & l_{23} \\ -1 & 1 & 0 & -l_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & l_{24} & 0 & 0 \end{bmatrix} \times \begin{bmatrix} M_1 \\ M_{21} \\ M_{24} \\ R_1^y \\ R_3^y \\ R_4^y \\ R_1^x \\ R_3^x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\frac{3}{8} q l_{24}^3 \\ 0 \\ q l_{24} \\ q l_{24} (l_{12} + \frac{1}{2} l_{24}) \\ 0 \\ \frac{1}{2} q l_{24}^2 \end{bmatrix}. \tag{21}$$

Finally, we take the values of lengths of the beams and load to be, respectively, $l_{12} = l_{24} = 1$, $l_{23} = 0.75$, and $q = 10$, but with the uncertainty of ±1%. Hence they can be represented by intervals:

$$l_{12} = l_{24} = [0.99, 1.01],\ \ l_{23} = [0.7425, 0.7575],\ \ q = [9.9, 10.1]. \tag{22}$$

We assume that there is no prestressing of the structure due to inexact dimensions of the beams. For that, we can either treat the uncertainty as the errors of measurements of the elements of the already existing structure, or else assume the structure will be assembled from inexact elements, but in a way that does not lead to prestressing (e.g., by slightly moving appropriate supports when necessary).

Substituting these values to Eq. (21) and calculating the interval coefficients according to the rules given by Eqs. (1, 2) we get finally the linear interval system with matrices:

$$\mathbf{A} = \begin{bmatrix} [1.98, 2.02] & [0.99, 1.01] & 0 & 0 & 0 & 0 & 0 & 0 \\ [0.99, 1.01] & [3.465, 3.535] & [-1.515,-1.485] & 0 & 0 & 0 & 0 & 0 \\ 0 & [-1.515,-1.485] & [4.455, 4.545] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & [0.99, 1.01] & [1.98, 2.02] & 0 & [0.7425, 0.7575] \\ -1 & 1 & 0 & [-1.01,-0.99] & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & [0.99, 1.01] & 0 & 0 \end{bmatrix}, \tag{23}$$

$$\mathbf{b} \approx \big(0,\ 0,\ [-3.9023,\ -3.6022],\ 0,\ [9.801,\ 10.201],\ [14.5545,\ 15.4545],\ 0,\ [4{,}8515,\ 5.1515]\big)^{\mathrm{T}}.$$

Solving the system with the methods described in Sec. 3, we obtained interval estimations of the solution set summarised in Table 2 below. The table includes the exact estimate (the hull) of the solution set as well as the estimates obtained by the approximate algorithms of Sec. 3.2. As it happened, both approximate methods tested by us produced the same estimations (within the accuracy used in the table). For comparison, the table contains also the *nominal solution* $\mathbf{x}_0$, i.e., the solution of the non-interval system of Eq. (21) for nominal values of parameters $l_{12} = l_{24} = 1$, $l_{23} = 0.75$, and $q = 10$, as well as the widths of the interval estimations relative to this nominal solution (in percent).

Table 2. Comparison of interval estimates for the example frame.

	$\mathbf{x}_0$	hull $\Sigma(\mathbf{A}, \mathbf{b})$	wid$[*]/\mathbf{x}_0$ [%]	PGSI, PGE	wid$[*]/\mathbf{x}_0$ [%]
M_1	0.25	[0.225, 0.278]	21.2	[0.223, 0.278]	22
M_{21}	–0.5	[–0.545, –0.459]	17.2	[–0.545, –0.457]	17.6
M_{24}	–1	[–1.061, –0.943]	11.9	[–1.061, –0.942]	12
R_1^y	–0.75	[–0.831, –0.677]	20.6	[–0.831, –0.671]	21.4
R_3^y	6.75	[6.227, 7.280]	15.6	[6.216, 7.285]	15.8
R_4^y	4	[3.753, 4.252]	12.5	[3.753, 4.254]	12.5
R_1^x	–0.667	[–2.059, 0.723]	417	[–2.088, 0.748]	425
R_3^x	0.667	[–0.723, 2.059]	417	[–0.748, 2.088]	423

As can be seen from the table, the resulting intervals are quite wide compared to only 2% width of the parameter intervals, though in general they are much narrower than for the truss example, seemingly due to much smaller widths of parameter intervals. The tightest possible estimate in this formulation, namely the hull of the solution set, produces the relative width at least 6 times greater than the width of the parameters, and for two unknowns (horizontal reactions R_1^x and R_3^x) as much as more than 200 times greater. Also the sign (i.e., direction) of estimates for these two reactions becomes uncertain (indicated by underlining). Both polynomial-complexity methods (PGSI and PGE) produced identical results, adding only a little overestimation over the hull. It confirms the observation made for the truss example in Sec. 4.1.2 that the overestimations for the hull of the solution set tend to grow fast with the widths of the

coefficients; for our frame, the widths are small (only 2% compared with 10% or 20% for the truss example).

Again, the sources of these overestimations, together with the possible approaches to reduce them, are discussed in the next Section.

5. PROBLEMS WITH INTERVAL APPROACH

The sometimes large widths of the solution intervals as compared with uncertainty of the parameters, occurring in the examples discussed in the previous Section, may raise doubts about practical usefulness of interval methods for analysis of linear mechanical systems. The answer to these doubts depends on the question whether the results reflect the real uncertainty of the solutions, or constitute only an artefact of the method itself. Fortunately, a large part of the overestimation is due to certain deficiencies in the formulation of the problem; in theory, with proper handling of interval arithmetics peculiarities, much better results can be obtained.

Any new technique or approach introduces new possibilities and promises of solving new kinds of problems, but it also introduces some new problems and difficulties of its own. Interval methods are no exception here. In this Section, several basic difficulties with application of interval methods to solving linear mechanical problems are identified and discussed. The discussion provides some advice as to applicability and relative merits of different approaches, as well as indicates several urgent and promising avenues for further research in this comparatively young field of numerical analysis.

5.1. Inexact estimates and algorithm efficiency

The estimates of solutions of interval systems of equations possible to obtain by the methods available today are usually not very accurate. There are several different types (and sources) of this inaccuracy, the main of them being:

- The methods produce interval enclosures (n-dimensional rectangular boxes) which are a rather bad approximations to the often star-like or elongated shapes of the solution sets. As a result, the real solution sets occupy often only a tiny percentage of the volume of the enclosing box, even for the tightest possible interval enclosures (hulls).
- More computationally efficient estimation methods produce enclosures often substantially larger than the tightest possible, in some cases with overestimations growing fast with the size of the system or the amount of uncertainty in the parameters.
- Even more fundamentally, the very formulation of linear mechanical problems in the form of linear interval systems of equations leads to large overestimations caused by certain properties of interval arithmetic, as mentioned in Sec. 2.1. For linear systems of equations the effect is known under the name of *coefficient dependence problem.*

The coefficient dependence problem is of different and more fundamental nature than the other two, which justifies its separate treatment in Sec. 5.2 below. The other two problems, together with the related question of efficiency (computational complexity) of algorithms for finding solution set enclosures, are discussed in the next two subsections.

5.1.1. Interval enclosures versus shapes of solution sets

Let us illustrate the problem with a two-dimensional example of an interval linear system whose solution set is a diagonal rectangle (Fig. 4). Depending on the ratio of widths of the coefficients b_1 and b_2, the solution set can be arbitrarily elongated, and thus the ratio of its area S_s to the area S_h of the smallest enclosing two-dimensional interval (hull) can be made arbitrarily small. In this case, $S_s / S_h = 2 \text{ wid } b_1 \text{ wid } b_2 \ / (\text{wid } b_1 + \text{wid } b_2)^2$. Hence the ratio goes to zero if any of the widths goes to zero,

with the maximal value of 1/2 attained for wid b_1 = wid b_2 (the solution set is then a square rotated by 45 degrees). As a result, in most cases the interval enclosure contains mostly points that *do not* belong to the solution set, and provides rather little information about the shape of the solution set.

Note that such elongated shapes of solution sets are not uncommon in mechanical structures we analysed. Moreover, the *direction* of elongation of the solution set of this kind may be a very useful and informative result in applications (see our other papers [16, 17]). Thus, another kind (shape) of the enclosure might be much more useful in such cases. One possible approach consists in finding, for a given linear interval system with solution set $\Sigma(\mathbf{A}', \mathbf{b}')$, another linear interval system of the type shown in Fig. 4, with a solution set $\Sigma(\mathbf{A}'', \mathbf{b}'')$ such that $\Sigma(\mathbf{A}', \mathbf{b}') \subseteq \Sigma(\mathbf{A}'', \mathbf{b}'')$. Then, the coefficients of the real matrix $\mathbf{A}''$ and interval vector $\mathbf{b}''$ would provide the parameters (orientation, position and dimensions) of an enclosing rectangular box in the $Ox_1 \ldots x_n$ space[4]. As far as the authors of this paper know, no work using this idea has been published. So, it remains a promising avenue of further research.

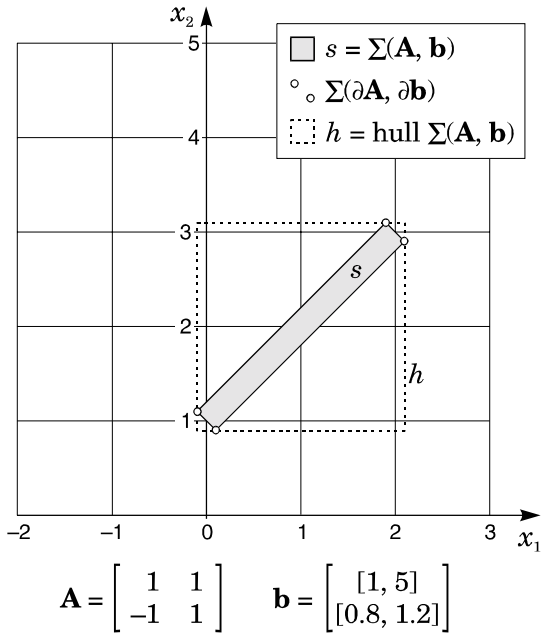


Fig. 4. An example of an “elongated rectangle” solution set.

In many other cases, the shape of the solution set is star-like with elongated spikes (rays) protruding to arbitrary distances. The spikes become longer and longer when the matrix of the system approaches a singular matrix. In the limit, when the matrix becomes singular, the end of the spike goes to infinity and the solution set becomes unbounded. It can be illustrated by the system shown in Fig. 5 below. For example, the left hand spike, currently at the point (–2, 4), may be shifted arbitrarily far away to the upper left by lowering the value of the coefficient $\underline{a}_{22}$ from –1 down to –2, where the singularity will appear. For such ill-conditioned systems some more informative characterisation of the solution set, other than the simple interval enclosure, would be much more useful. As these phenomena occur mostly for the systems near singularity, investigation of singular and near-singular interval systems should be useful here (see Sec. 5.3 below for additional discussion of problems with singularity treatment).

Partial answer to such problems may be in some cases provided by calculating also *inclusion-maximal inner interval estimates*, i.e. multidimensional intervals (boxes) wholly *included* in the solution set [32,

[4] Systems with real (non-singular) matrix $\mathbf{A}$ and intervals occurring only in the right-hand side vector $\mathbf{b}$ produce solutions sets that are parallelepipeds; when $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ is a diagonal matrix they became rectangular prisms. For orthogonal matrices (i.e., when $\mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{I}$), the parameters of the prism are separated in the system—rows of $\mathbf{A}$ become simply versors for the directions of the prism edges, while intervals in $\mathbf{b}$ give positions and lengths of the edges along the directions given by the versors.

33]. However, in many cases (like the one with "elongated rectangle" solution set discussed above), the maximal inner estimate also does not carry much information about the shape of the solution set and may be even not unique. Quite another possibility is to approximate the solution set not by a single interval bounding box, but by a number of smaller boxes. This technique does not solve all the problems, but is useful in some applications, especially in optimisation [8].

5.1.2. Inexact and inefficient estimation algorithms

In addition to the kind of overestimation due to the sometimes awkward shapes of solution sets, the methods producing interval enclosures of the set may produce additional overestimation. Moreover, the overestimation problem is closely linked with computational complexity of estimation algorithms. Namely, as it was mentioned in Sec. 3, the methods that produce exact hull of the solution set are of exponential complexity, hence not very practical for larger systems of equations, while the polynomial complexity algorithms produce the result faster, but for the price of often large overestimation of the hull.

One cannot expect any significant improvement of the situation, as it was proved that finding the exact hull is a *NP*-hard problem[5)] [30]. Later it was proven that even computing an enclosure with relative overestimation $\Delta\overline{x}_i \,/\, \overline{x}_i$ (for each $\overline{x}_i \neq 0$) smaller than $4/n^2$ and with absolute overestimation $\Delta\overline{x}_i$ smaller than $1/4n^4$ is *NP*-hard too [28]. Note that in the above formulas $\overline{x}_i$ is the upper bound of the i-th component of the interval enclosure vector, not its width. It seems the above estimates of "attainable polynomial accuracy" are rather conservative on the low side, as they are decreasing rather fast with the size n of the system, contrary to the behaviour of most known estimation algorithms which produce overestimations that *grow* with the size n.

The polynomial-complexity estimation algorithms produce interval bounds that overestimate the exact enclosure (hull). The amount of overestimation varies between algorithms and usually grows with the size n of the system of equations and/or with the norm of the right-hand side vector $\mathbf{b}$, sometimes quite fast [8, 20]. However, in many cases the bounds on overestimation are not known or are very crude, giving only little information on the possible amount of estimation error.

Also, the general indication of the type of growth of the overestimation is not very useful in practical applications where we are rather interested in knowing the size of estimation error for a particular interval system at hand, not the general behaviour of the algorithm for all conceivable problems. In this respect, a very promising approach is offered by algorithms that provide lower and upper bounds for the hull, like the Rump algorithms [31], discussed in Sec. 5.2.2 below in connection with the coefficient dependency problem. Even if such bounds are crude, they still provide more information of practical importance than the general indication of the type of growth of the estimation error. Hence, at least from the practical point of view, investigation of the possibility to obtain such bounds from existing algorithms, or finding new algorithms offering such a possibility, constitute another very useful direction of further work in the field.

5.1.3. Generality versus specialisation of estimation algorithms

Many interval methods for solving linear systems of equations are specialised for specific types of systems (that is, system matrices). For other types of matrices they either do not work (giving wrong results) or are far less effective (either with respect to computational complexity, or estimation accuracy, or both) than for the special case. The effects of this state of affairs are both positive and negative.

On the one hand, one can try to find a method for the particular problem at hand that may be much better than some general method, producing better results faster. For example, for systems with inverse positive matrices there is an algorithm (due to Beeck [2]) producing the exact interval enclosure (hull)

[5)] To be precise, they proved that it is *NP*-complete, i.e. of the same complexity as problems from the *NP* complexity class which are believed to be of exponential complexity, though at the current stage of development of the complexity theory that conjecture has not yet been proved.

with *linear* computational complexity—requiring only to solve at most $2n$ real systems of equations (with selected boundary matrices).

On the other hand, testing applicability of a given method to the given case is often quite involved and computationally expensive (see e.g. the applicability condition of the singular values calculation method for interval matrices given in [5]). This additional overhead may easily neutralise the gain expected from the specialised method. This is often aggravated by the fact that the classification of matrices implied by various algorithms often does not necessarily coincide with the classification that is meaningful from the application point of view. Hence, the method painfully tested to be applicable to one problem may happen to be not applicable to a problem quite similar from the application point of view (not speaking about finding in the still scattered and unstructured literature that single method which may be good for the given case...).

As the field is still rather young, all these causes are still active in producing substantial amount of confusion. That certainly calls for some tidying and structuring work, possibly culminating in production of a compendium book serving as an interval counterpart of the highly popular "*Numerical Recipes*" [24], or even some automatic or semi-automatic tools serving as consultants for those trying to apply interval methods to real-world problems.

5.2. Interval arithmetic deficiencies and coefficient dependence

As it was hinted at in Sec. 2.1, certain formal deficiencies of the interval arithmetic, like that the usual distributive law is not valid here, only the weaker subdistributive law of Eq. (3), may lead to overestimation errors during calculation of interval expressions. Indeed, if not taken properly into account, these effects may severely diminish the accuracy of interval estimates of solutions of linear systems of even moderate complexity. In this Section, we attempt to explain the effects in more detail, show how they affect the accuracy of interval estimations, and discuss possible remedies to the problem.

5.2.1. Computing interval expressions

One of the unpleasant consequences of the mentioned weaknesses of interval arithmetic is that ordinary algebraic calculation of interval expressions in which some interval variable(s) occur more than once may produce wrong results (as compared to proper results given by Eqs. (1) or (1′); unfortunately, these formulas are rather hard to apply directly—see the discussion of the SPI method in Sec. 5.2.2 below). Only in certain cases such a calculation gives the valid result, otherwise substantial overestimations can arise [1, 8, 20]. Fortunately, due to inclusion isotonicity property (Eq. (4)), we know that the result will be always at most an overestimation—we never obtain an interval smaller than the correct result (see [8] for more precise conditions for that to be true). Hence, we shall always get a reliable upper bound(s) on the correct interval solution.

For example, consider the expression $f(x) = 1/(1+1/x)$, for $x \neq 0$. In standard arithmetic it is equivalent to a simpler formula $g(x) = x/(x+1)$, requiring only one division instead of two required for $f(x)$, but containing the variable x twice. Putting $x = [2, 3]$ and using directly the interval arithmetic rules given by Eq. (2), one gets:

$f([2, 3]) = 1/(1+1/[2, 3]) = 1/(1+[1/3, 1/2]) = 1/[4/3, 3/2] = [2/3, 3/4]$,

(which is correct), whereas:

$g([2, 3]) = [2, 3]/([2, 3]+1) = [2, 3]/[3, 4] = [2, 3] \cdot [1/4, 1/3] = [1/2, 1]$.

Thus, $g([2, 3]) \neq f([2, 3])$, and calculation according to the seemingly simpler formula $g(x)$ overestimates significantly the value of $f(x)$. The problem is that many functions cannot be expressed in the form in which the variables occur only once (or in certain other forms for which the overestimation problem does not arise too [1]). For another example, take an even simpler problem of calculating x^2. Putting $x = [-1, 1]$ and using the general rule defining interval arithmetic operators given by Eq. (1), we get, quite naturally, $[-1, 1]^2 = [0, 1]$, while using the formula $x^2 = x \cdot x$, obvious in arithmetic of reals, one gets, quite wrongly, $[-1, 1]^2 = [-1, 1] \cdot [-1, 1] = [-1, 1]$. Thus formulas of the sort $x^2 + x$ and

$x\,(x+1)$ are not only not equivalent in interval arithmetic, but also cannot be transformed into forms giving correct values when evaluated with straightforward application of interval arithmetic operations. Namely, with standard arithmetic rules, we have:

$$(x^2 + x)\,\big|_{\,x\,:=\,[-1,\,1]} = [-1,\,1]^2 + [-1,\,1] = [0,\,1] + [-1,\,1] = [-1,\,2],$$

$$(x\,(x+1))\,\big|_{\,x\,:=\,[-1,\,1]} = [-1,\,1]\,([-1,\,1] + 1) = [-1,\,1]\,[0,\,2] = [-2,\,2],$$

while using the basic definition of interval operation extension (Eq. (1)) we get the single correct result:

$$\{\tilde{x}^2 + \tilde{x} \mid \tilde{x} \in [-1,\,1]\} = \{\tilde{x}\,(\tilde{x} + 1) \mid \tilde{x} \in [-1,\,1]\} = [-1/4,\,2].$$

5.2.2. Matrix coefficients dependence and parametric formulation

In the case of systems of linear interval equations originating from problems of mechanics, this leads to the so-called *coefficient dependence problem*. For these systems, the coefficients in the equations are not independent quantities—they are all functions of some (usually few) physical system parameters (e.g., stiffnesses of the bars in truss structures). Thus, the problem is in fact of the form:

$$\mathbf{A}(\mathbf{p})\,\mathbf{x} = \mathbf{b}(\mathbf{p}), \quad \text{with } \mathbf{p} = (p_1, p_2, \ldots, p_k), \tag{24}$$

where p_i, $i = 1, \ldots, k$, are given parameters varying over specified intervals and for every $\tilde{\mathbf{p}} \in \mathbf{p}$, $\mathbf{A}(\tilde{\mathbf{p}})$ and $\mathbf{b}(\tilde{\mathbf{p}})$ are real matrices. Since, usually, many different coefficients of the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ depend on the same parameters, the variability of the coefficients within their intervals is no longer independent (as is assumed in the classical formulation). Hence, in the resulting expression for the solution $\mathbf{x}$ the same parameter occurs several times. Note also that the α-cut method for solving fuzzy equations [3] also leads naturally to parametric interval equations, with the threshold value $\alpha \in [0, 1]$ acting as an interval parameter.

The set of solutions of the problem formulated by Eq. (24) is naturally defined as:

$$\Sigma(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p})) = \{\tilde{\mathbf{x}} \in \mathsf{R}^n \mid (\exists\, \tilde{\mathbf{p}} \in \mathbf{p})\; \mathbf{A}(\tilde{\mathbf{p}})\,\tilde{\mathbf{x}} = \mathbf{b}(\tilde{\mathbf{p}})\}. \tag{25}$$

Due to the properties of interval arithmetic explained in the previous subsection, solving such interval systems with interval arithmetic without taking the dependency into account produces usually much larger solution set than the true one defined by Eq. (25).

An attempt to calculate the hull of the solution set of Eq. (25) with the CEIC method, i.e. by solving 2^k real-number systems of equations for all combinations of endpoints of the interval parameters p_i (that is, for all members of $\partial\mathbf{p}$), will not produce valid results unless it is assured that components of the solution vector $\mathbf{x}$ depend monotonically upon all p_i 's. This is rarely the case: as is shown by the simple two-dimensional example of Fig. 5, even when some coefficients of the system depend linearly and monotonically on a single interval parameter, the solutions depend on the parameter $\mathbf{p}$ non-monotonically, hence in general we have only the relation of inclusion here:

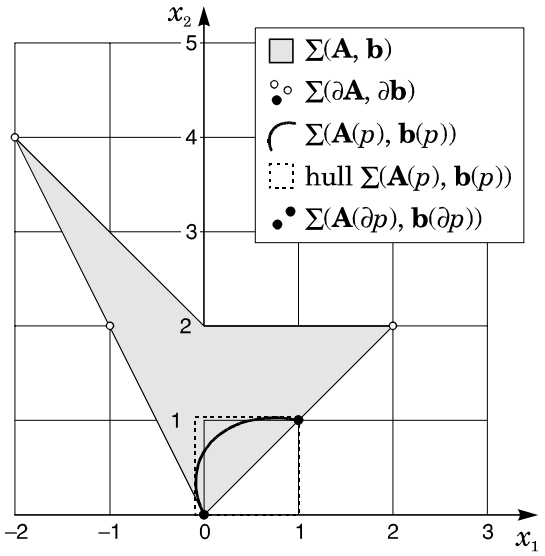
$$\text{hull}\,\Sigma(\mathbf{A}(\partial\mathbf{p}), \mathbf{b}(\partial\mathbf{p})) \subseteq \text{hull}\,\Sigma(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p})),$$

not necessarily an equality. For instance, in the example of Fig. 5 we have:

$$\text{hull}\,\Sigma(\mathbf{A}(p), \mathbf{b}(p)) = \begin{bmatrix} [-2/23,\,1] \\ [0,\,4/(2\sqrt{6}-1)] \end{bmatrix} \approx \begin{bmatrix} [-0.087,\,1] \\ [0,\,1.026] \end{bmatrix},$$

while:

$$\text{hull}\,\Sigma(\mathbf{A}(\partial p), \mathbf{b}(\partial p)) = \begin{bmatrix} [0,\,1] \\ [0,\,1] \end{bmatrix} \neq \text{hull}\,\Sigma(\mathbf{A}(p), \mathbf{b}(p)).$$



$$\mathbf{A} = \begin{bmatrix} [0, 1] & 1 \\ -2 & [-1, 2] \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} [0, 2] \\ 0 \end{bmatrix},$$

$$\mathbf{A}(p) = \begin{bmatrix} p & 1 \\ -2 & 3p-1 \end{bmatrix} \quad \mathbf{b}(p) = \begin{bmatrix} 2p \\ 0 \end{bmatrix}, \; p \in [0, 1]$$

Fig. 5. A parametric system example.

The comparison of hull $\Sigma(\mathbf{A}(p), \mathbf{b}(p)) = ([-0.087, 1], [0, 1.026])^{\mathrm{T}}$ with hull $\Sigma(\mathbf{A}, \mathbf{b}) = ([-2, 2], [0, 4])^{\mathrm{T}}$ shows also the magnitude of overestimation caused by not taking into account the coefficient dependence. Worse still, such an overestimation can be arbitrarily large. For instance, the upper-left extremal point of $\Sigma(\mathbf{A}(p), \mathbf{b}(p))$ may be shifted arbitrarily far away to the upper left by simply lowering the value of the left endpoint $\underline{a}_{22}$ of the coefficient a_{22} from -1 down to -2 (which would amount to changing its parametric form from $3p-1$ up to $4p-2$). As can be easily shown, hull $\Sigma(\mathbf{A}(p), \mathbf{b}(p))$ will grow only a little during this process, hence the overestimation will grow indefinitely. At the limit (for $\underline{a}_{22} = -2$), the matrix $\mathbf{A}$ becomes singular, $\Sigma(\mathbf{A}, \mathbf{b})$ becomes unbounded, but $\Sigma(\mathbf{A}(p), \mathbf{b}(p))$ remains still bounded and small (namely, for $\underline{a}_{22} = -2$ the hull of this set will equal approximately $([-0.286, 1], [0, 1.094])^{\mathrm{T}}$). Hence, from an initially non-singular system (obtained when the coefficient dependence is properly taken into account), ignoring the dependence may lead not only to an overestimation of the solution set, but even to a singular system with an unbounded solution set (see Sec. 5.3 below for further examples and discussion of this subject).

The brute-force method for obtaining hull $\Sigma(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p}))$ would be thus to compute it directly according to Eq. (1) (or rather Eq. (1′)), i.e., to calculate the set of solutions to the real system $\mathbf{A}(\tilde{\mathbf{p}})\tilde{\mathbf{x}} = \mathbf{b}(\tilde{\mathbf{p}})$ for all combinations of values of parameters $\mathbf{p}$ and take its lower and upper bounds. Unfortunately, in practice it is impossible, as the number of combinations of parameter values is infinite. What we can do is only to sample this set of combinations systematically, with hope that our samples will fall at (or at least sufficiently near to) the values giving extremal solutions of the system. Such a method, which can be called *sampling of parameter intervals* (*SPI*) is certainly very inefficient, usable only for very small number of parameters, and of doubtful accuracy, as there is no guarantee of hitting the exact extremal point of the function $\tilde{\mathbf{x}}(\mathbf{p})$. However, if this function is continuous, with decreasing sampling step we may approach the exact solution with an arbitrary accuracy (if only our computing resources would allow). Which may be still practical for assessing qualitatively the order of

overestimation of the solution produced by coefficient dependence, especially when the number of parameters is small.

There are only a few results on solving the general problem of finding interval estimates for functions, or linear interval systems of equations, depending on interval parameters. Jansson [10] describes the method working with symmetry constraint, i.e., giving interval estimates for systems with symmetric interval matrix $\mathbf{A}$ such that all real matrices considered in the definition of the solution set are also symmetric. That is, it gives an (outer) interval estimate for:

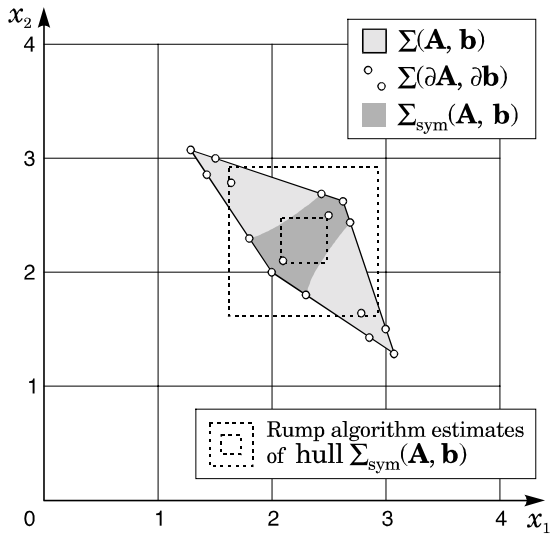
$$\sum_{\mathrm{sym}}(\mathbf{A}, \mathbf{b}) = \{\tilde{\mathbf{x}} \in \mathrm{R}^n \mid \mathbf{A} = \mathbf{A}^{\mathrm{T}} \,\&\, (\exists\, \tilde{\mathbf{A}} \in \mathbf{A})\,(\exists \tilde{\mathbf{b}} \in \mathbf{b})\; \tilde{\mathbf{A}} = \tilde{\mathbf{A}}^{\mathrm{T}} \,\&\, \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}\}.$$

Rump [31] generalised the method for any linear dependence of the system coefficients on finite number of interval parameters. His method gives additionally an estimate of accuracy of the obtained results, as it produces two estimates: a lower bound $\mathbf{x}^-$ and an upper bound $\mathbf{x}^+$ such that:

$$\mathbf{x}^- \subseteq \mathrm{hull} \sum_{\mathrm{lin}}(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p})) \subseteq \mathbf{x}^+, \text{ with } \mathbf{p} = (p_1, p_2, \ldots, p_k).$$

The above can be illustrated by a simple two-dimensional example used by Rump [31] (originated by Behnke). The interval matrices (original one and that in a parametric form producing a symmetric matrix) and various solution sets of the above system, together with estimates of the symmetric solution set produced by the Rump algorithm, are shown in Fig. 6. Note that he solution set for the symmetric system is not a polygon (two of its sides are curved). The estimates produced by the Rump algorithm are rather crude—the errors relative to the width of $\mathrm{hull} \sum_{\mathrm{sym}}(\mathbf{A}, \mathbf{b})$ fall between around 20% and more than 30%.

Another idea is to use global interval optimisation technique [8] to find the bounds on $\sum(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p}))$. However, it needs still considerable research to become practical for this kind of objective function.



$$\mathbf{A} = \begin{bmatrix} 3 & [1, 2] \\ [1, 2] & 3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} [10, 10.5] \\ [10, 10.5] \end{bmatrix}$$

$$\mathbf{A}_{\mathrm{sym}}(p) = \begin{bmatrix} 3 & p \\ p & 3 \end{bmatrix}, \; p \in [1, 2]$$

Fig. 6. A Rump-Behnke example.

5.2.3. Coefficients dependence effects in linear mechanical systems

The dependence of coefficients of the system of equations on a set of parameters is a common phenomenon in mechanical systems. Also, in most cases there is much smaller number of interval parameters than the number of coefficients of the system, and many coefficients may depend on a single parameter. Not surprisingly, not taking this dependence into account leads to large overestimations of the interval solutions of the system, often too large to be practical in real-life mechanical applications (see discussion of the examples in Sec. 4.1). From our experience it seems to be the single most harmful feature constraining the use of interval methods for analysis of real mechanical systems with uncertainties. Hence, the investigation of this problem is possibly the most important necessity from the applications point of view.

What can be gained in estimation accuracy when the coefficients dependence is properly handled can be shown on some of our mechanical examples.

5.2.3.1. Coefficients dependence for the truss example

To assess the amount of overestimation caused by coefficient dependence for our truss example of Fig. 2 we may use the brute-force SPI method described above in Sec. 5.2.2. It can be successfully applied in this case as the size of the system of equations and the number of parameters is small (only one parameter in two cases and two parameters in the third one). The hulls of the solution set for the parametric formulation for this example are summarised in Table 3 below.

Table 3. Hull $\square(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p}))$: Interval results for the parametric formulation of the example truss problem (with 7×7 matrix, all three uncertainty cases).

	$\mathbf{d}_0$ [$\times 10^{-3}$]	*a1a* [s_{23} ±5%] [$\times 10^{-3}$]	wid$[*]/\mathbf{d}_0$ [%]	*a1* [s_{23} ±10%] [$\times 10^{-3}$]	wid$[*]/\mathbf{d}_0$ [%]	*a2* [s_{23}, s_{34} ±5%] [$\times 10^{-3}$]	wid$[*]/\mathbf{d}_0$ [%]
d_1^x	–20	–0.02	0	–0.02	0	–0.02	0
d_2^x	–2.5	[–2.58, –2.41]	7.1	[–2.66, –2.30]	14.3	[–2.67, –2.31]	14.2
d_2^y	–38.71	[–38.81, –38.63]	0.5	[–38.91, –38.55]	0.9	[–38.90, –38.54]	0.9
d_3^x	–5.0	–5.0	0	–5.0	0	–5.0	0
d_3^y	–34.14	[–34.31, –33.96]	1.0	[–34.46, –33.75]	2.1	[–34.50, –33.79]	2.1
d_4^x	–12.5	[–12.58, –12.41]	1.4	[–12.66, –12.30]	2.9	[–12.67, –12.31]	2.8
d_4^y	–19.57	[–19.66, –19.48]	0.9	[–19.73, –19.37]	1.8	[–19.74, –19.39]	1.8

As can be seen from the table, the interval results are quite tight around the nominal solution $\mathbf{d}_0$. They are so small that they even cannot be adequately represented in the scale of Fig. 2b. The relative width of the displacement intervals is smaller than the width of the parameter interval—a situation physically understandable as the effects of uncertainty of a single local parameter of the truss get distributed around the structure. For certain displacements we can even see that they are not affected by the parameter uncertainty at all.

Comparison of the above results with those in Table 1 in Sec. 4.1.2 shows how much can be gained in the accuracy of interval estimation of solutions of equations for mechanical structures when the coefficient dependence is properly taken into account. The most dramatic effect is with the *a2* case (two uncertain parameters). While in the nonparametric formulation the set of solutions became unbounded for this case (hence, practically nothing useful could be said about the uncertainty of the results), here the uncertainties are shown to be in fact quite small. The dependence of the results on the character of the uncertainty is also more well-behaved: the displacement uncertainty for the doubled parameter uncertainty is also doubled; similarly, for ±5% uncertainty of two parameters the results are almost the same as for ±10% uncertainty of one parameter.

5.2.3.2. Coefficients dependence for the frame example

The frame example of Sec. 4.2 leads to the system of equations given by Eq. (21). It can be easily solved analytically, resulting in the following formulas:

$$\begin{aligned}
M_1 &= \tfrac{1}{4} C l_{24}^2 l_{23},\\
M_{21} &= -2M_1,\\
M_{24} &= -\tfrac{1}{8} C l_{24}^2 (3l_{12} + 4l_{23}),\\
R_1^y &= -\tfrac{3}{4} C l_{24}^2 l_{23} / l_{12},\\
R_3^y &= \tfrac{1}{4} C (3l_{24}^2 l_{23} / l_{12} - l_{12} l_{23}) + \tfrac{5}{8} q l_{24},\\
R_4^y &= \tfrac{1}{4} C l_{12} l_{23} + \tfrac{3}{8} q l_{24},\\
R_1^x &= -\tfrac{3}{8} C l_{24}^2 l_{12} / l_{23},\\
R_3^x &= -R_1^x,
\end{aligned} \tag{26}$$

where:

$$C = \frac{q l_{24}}{3l_{12} l_{24} + 2l_{12} l_{23} + 4l_{24} l_{23}}. \tag{26′}$$

Substituting in the obtained formulas the interval values of parameters from Eq. (22) and calculating the interval results according to the proper rules given by Eq. (1), we get the *exact* interval solution to the problem, given by $\mathbf{x}_E = \text{hull}\ \sum(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p}))$, where $\mathbf{p} = (l_{12}, l_{24}, l_{23}, q)$. For comparison, we also calculate the interval results by applying straightforwardly the rules of Eq. (2), thus obtaining the interval solution $\mathbf{x}_I$. Obviously, due to isotonicity, we must have $\mathbf{x}_E \subseteq \mathbf{x}_I$. The solutions are given in Table 4, together with the nominal solution $\mathbf{x}_0$ and the width of the interval results relative to the nominal solution, similarly as in Table 2 in Sec. 4.2.2. Let us recall that the relative width of the parameter intervals equals 2%.

Table 4. Interval results for analytical solution of the example frame.

	$\mathbf{x}_0$	$\mathbf{x}_E$	wid $\mathbf{x}_E/\mathbf{x}_0$ [%]	$\mathbf{x}_I$	wid $\mathbf{x}_E/\mathbf{x}_0$ [%]
M_1	0.25	[0.240, 0.261]	8.4	[0.233, 0.268]	14
M_{21}	–0.5	[-0.521, -0.479]	8.4	[-0.536, -0.466]	14
M_{24}	–1	[-1.034, -0.966]	6.8	[-1.072, -0.932]	14
R_1^y	–0.75	[-0.790, -0.712]	10.4	[-0.812, -0.692]	16
R_3^y	6.75	[6.591, 6.913]	4.8	[6.573, 6.933]	5.3
R_4^y	4	[3.920, 4.080]	4	[3.911, 4.091]	4.5
R_1^x	–0.667	[-0.702, -0.633]	10.4	[-0.722, -0.615]	16
R_3^x	0.667	[0.633, 0.702]	10.4	[0.615,0.722]	16

Comparing solutions $\mathbf{x}_E$ and $\mathbf{x}_I$ we can already see the marked overestimation effects (causing significant widening of the estimation intervals, e.g. more than two times, relatively, for M_{24}) that may result from evaluation of even quite simple interval expressions. Note that different form of formulas in

Eq. (26) may lead to another results of their interval evaluation. For instance, when the factor C from Eq. (26′) is substituted to Eq. (26) and the resulting formulas are transformed to a single fraction form, the overestimations of interval evaluation $\mathbf{x}_\mathrm{I}$ for reactions R_3^y and R_4^y rise to 16% and 12% respectively (other results remaining the same).

Also, comparing Table 4 with Table 2, we can again clearly see the magnitude of overestimation that may occur when the coefficient dependencies in interval systems of equations are not handled properly, even for such a comparatively simple system. Note also that the overestimations due to this effect may be quite different in magnitude for different variables (compare overestimation percentages for R_1^x and R_3^x in Table 2 with values for other unknowns). Interestingly, although for most unknowns hull $\boldsymbol{\Sigma}(\mathbf{A}, \mathbf{b})$ is wider than $\mathbf{x}_\mathrm{I}$ (as could be expected), for M_{24} actually $\mathbf{x}_\mathrm{I}$ is wider. Obviously, $\mathbf{x}_\mathrm{E} \subseteq$ hull $\boldsymbol{\Sigma}(\mathbf{A}, \mathbf{b})$ always.

5.2.3.3. Solving coefficients dependence problem for mechanical structures

As presented examples show, various types of dependencies can occur in matrices of mechanical systems. For some systems, we get symmetric matrices (like for trusses, see Sec. 4.1.1). For others, the dependence on many important design parameters is linear (like on s_{ij}, A_{ij}, and E_{ij} for trusses), or is linear for some convenient function of the design parameter (e.g., for trusses the dependence on l_{ij} is non-linear, but it is linear on $1/l_{ij}$). Hence, the methods, described in the previous subsection, of tackling such types of dependence constitute a promising starting point for the development of useful algorithms.

However, their practical usefulness in current state of the art is still limited. First, let us note that for truss structures the constraint of symmetry of the matrix does not capture all the dependence that there exists. Among others, while properly representing the equalities $a_{ij}^{yx} = a_{ji}^{xy}$ and $a_{ij}^{xy} = a_{ji}^{yx}$, it does not capture the fact that also $a_{ij}^{xy} = a_{ij}^{yx}$, see Eqs. (19, 20). Second, while the dependency is linear on some parameters, say A_{ij} and E_{ij}, individually, in the case when both parameters are uncertain and should be varied independently over their separate intervals, the dependency becomes non-linear. Also, dependence on some design parameters is inherently non-linear, like, for trusses, dependence on bar and external load directions (angles). Similar observations can be made for other mechanical structures, e.g. frames (see Sec. 4.2).

The above mentioned methods for handling coefficient dependency were not yet fully tested by us on real mechanical problems. Regardless of that, because of the known limitations of the discussed methods, we may safely say that further research into this area will be needed anyway. It is currently carried out by our group, for the particular case of analysis of linear mechanical structures, and the results will be reported in a separate publication.

5.3. Inadequate development of singular systems theory and methods

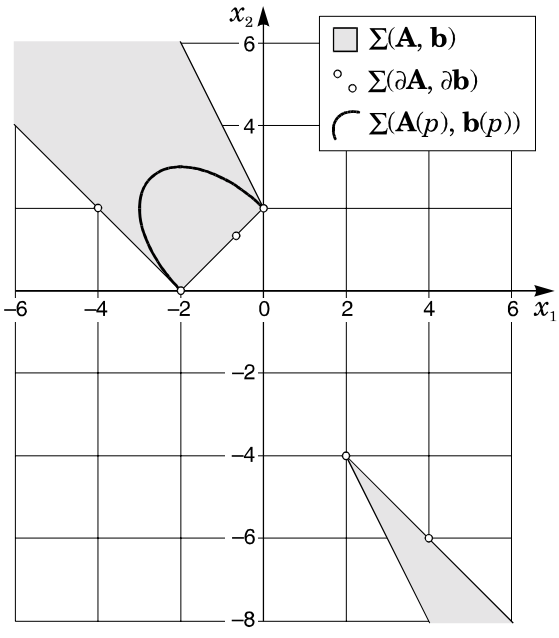
The overwhelming majority of results in solving linear interval systems of equations concerns non-singular systems. Singular systems were mostly left out as (probably) uninteresting. At the most, methods for testing for singularity of interval matrices were investigated to some extent. However, in interval formulation the importance of singular systems is much greater than in real matrix theory and applications.

First, the very process of formulating the interval form of some real-world (e.g., mechanical) problem may lead to a singular interval system for a quite well-behaved initial problem (see the results for the *a2* case of the example truss discussed in Sec. 4.1.2 and 5.2.3.1), just because of acknowledging a somewhat larger uncertainty in some parameter (recall how moving one end of a coefficient interval [–1, 2] just from –1 to –2 in the example of Fig. 5 had led to a singular system), or due to the coefficients dependence effects discussed above [8]. Here, of considerable interest is, e.g., to be able to find what changes in parameter uncertainties should be made to avoid producing a singular matrix, or, more generally, what is the position and “shape” of the set of singular matrices within the given matrix interval.

Some sort of an interval extension of singular value decomposition (*SVD*) methods developed in real matrix theory may be of great interest here. Again, only some preliminary steps in this direction have as yet been made [5].

Second, the interval matrix being singular means usually that only *some* of the real matrices included in it are singular—in most of the interesting cases the majority of them are non-singular. It means, for example, that due to the singularity of the system the general set of solutions $\Sigma(\mathbf{A}, \mathbf{b})$ may be unbounded (an extreme of overestimation!), while the parametric set $\Sigma(\mathbf{A}(\mathbf{p}), \mathbf{b}(\mathbf{p}))$ that we may be far more interested in, can quite well be bounded (again, compare the results for the *a2* case of the example truss of Sec. 4.1.2 and 5.2.3.1, and see an example in Fig. 7). Hence, the methods of handling of certain types of solution sets also for singular matrices would be very useful in practice.

Third, since for even non-singular interval systems we already get (usually infinite) *sets* of solutions, the boundary between sets of solutions for non-singular and singular systems is not so clear-cut as for non-interval systems—the difference is only that between bounded and unbounded sets. Even if unbounded, the shape of the set of solutions to a singular system may be of much interest, especially in qualitative analysis applications [6, 7, 11, 13]. For instance, the information that components of the solutions to the singular system shown in Fig. 7 are of opposite sign may be very important in such a kind of application. Hence, methods of estimating shapes of solutions sets for singular systems may be quite useful in practice. As far as the authors of this paper know, practically nothing has been done in this direction as yet.



$$\mathbf{A} = \begin{bmatrix} [0, 2] & 1 \\ -1 & [-1, 1] \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} [0, 2] \\ 2 \end{bmatrix},$$

$$\mathbf{A}(p) = \begin{bmatrix} 2p & 1 \\ -1 & 2p-1 \end{bmatrix} \quad \mathbf{b}(p) = \begin{bmatrix} 2p \\ 2 \end{bmatrix}, \; p \in [0, 1]$$

Fig. 7. A solution set of an example singular system.

It seems that for analysis of singular interval systems the extension of interval arithmetic called *Kahan arithmetic* [1, 18] may prove very useful. The extension considers also unbounded intervals and "outer" intervals. This would also add another possibility to the analysis of systems with uncertainties, as it allows for specification of the condition that some parameter lies *outside* a specified range, which is hard to specify with standard interval arithmetic. Some elements of this extension have already been successfully applied by Hansen to improve efficiency of iterative interval algorithms [8].

6. CONCLUSIONS

In the paper, we have investigated possibilities of and problems with application of interval methods in (qualitative) analysis of linear mechanical systems with parameter uncertainties. After giving an introduction to interval arithmetic, systems of linear interval equations, and basic methods for finding interval estimates for solutions of such systems, we have shown some examples of applying interval methods to practical problems of analysis of mechanical structures.

The results, on the one hand, suggest the usefulness and comparative simplicity of interval methods as applied to finding estimates of solutions to systems with parameter uncertainties, and, on the other hand, indicate that the methods must be used with proper caution. The most important lesson is that straightforward use of interval methods as if they were essentially identical to the arithmetic of reals, only with a little extended kind of "interval numbers", may lead to considerable, and cumulative, overestimation of the proper results. Although, fortunately, the results are reliable (or safe) in the sense that the final interval result is guaranteed to *include* the proper set of solutions (hence, no solutions are missed in this way), the overestimations may still amount to such a loss of information as to render the result practically useless.

For linear systems of equations the effect is called the *coefficient dependence problem* and provides the main source of overestimation of the results, as discussed in Sec. 5.2.2 and 5.2.3. Hence, proper formulation of the problem in interval form and careful application of interval arithmetic laws, taking into account their significant deviations from the familiar arithmetic on reals, is needed to alleviate these effects. There are also other sources of interval estimation inaccuracy, as discussed in more detail in Sec. 5.1.

Another problem with interval linear methods identified in the paper (Sec. 5.3) is concerned with far inadequate development of theory and methods of analysis of *singular* interval systems. This is especially serious because in interval formulation the practical importance of singular systems is much greater than in applications of standard linear systems.

The examples and following discussions have led to identification and formulation (in Sec. 5 of the paper) of several important and promising avenues for further research, both of purely theoretical nature, and involving construction and testing of new practical interval methods and algorithms. Advances in the indicated research areas will considerably boost the usefulness of interval methods in the field of analysis of linear mechanical systems with uncertainties.

ACKNOWLEDGEMENTS

The research reported here was supported by grant No. 8 T11F 006 08, and in part by grant No. 8 T11F 008 10, both from KBN (The State Committee for Scientific Research). Thanks are due to Mr. Aleksander Radomski for his help with the implementation of the methods. The encouragement and useful advice of Prof. Michað Kleiber is also greatly appreciated, as is the help in finding some literature references and useful discussions on various aspects of interval methods, kindly provided by members of the *Reliable-Computing* discussion list on the Internet, especially Prof. JiØÍ Rohn and Prof. Arnold Neumaier.

REFERENCES

[1] G. Alefeld, J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York 1983.

[2] H. Beeck. Zur Problematik der HÜllenbestimmung von Intervallgleichungssystemen. In: K. Nickel, ed., *Interval Mathematics 1975*. Lecture Notes in Computer Science, vol. **29**: 150-159. Springer Verlag, Berlin 1975.

[3] J.J. Buckley, Y. Qu. On using α-cuts to evaluate fuzzy equations. *Fuzzy Sets and Systems* **38**: 309-312, 1990.

[4] Z. Cywiñski. *Mechanika budowli w zadaniach.* [*Exercises in Structural Mechanics*, in Polish]. Polish Scientific Publishers (PWN), Warszawa 1984.

[5] A.S. Deif. Singular values of an interval matrix. *Lin. Alg. Appl.* **151**: 125-133, 1991.

[6] K.D. Forbus. Qualitative physics: past, present, and future. In: H.E. Shrobe, ed., *Exploring Artificial Intelligence*, 239–296. Morgan Kaufmann, San Mateo, CA, 1988.

[7] A. Gatnar. *Metody modelowania jakoöciowego.* [*Methods of Qualitative Modelling*, in Polish]. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.
[8] E. Hansen. *Global Optimization Using Interval Analysis.* Marcel Dekker, New York 1992.
[9] D.J. Hartfiel. Concerning the solution set of $Ax = b$ where $P \le A \le Q$ and $p \le b \le q$. *Numerische Mathematik*, **35**: 355-359, 1980.
[10] C. Jansson. Interval linear systems with symmetric matrices, skew-symmetric matrices and dependencies in the right hand side. *Computing*, **46**: 265-274, 1991.
[11] M. Kleiber, Z. Kulpa. Computer-assisted hybrid reasoning in simulation and analysis of physical systems. *CAMES*, **2**(3): 165-186, 1995.
[PostScript file: see http://www.ippt.gov.pl/~zkulpa/zkpubl.html#quaphys]
[12] H.U. KÖylÜoğlu, A.ª. Çakmak, S.R.K. Nielsen. Interval mapping in structural mechanics. In: Spanos, ed. *Computational Stochastic Mechanics*. 125-133. Balkema, Rotterdam 1995.
[13] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. The MIT Press, Cambridge, MA, 1994.
[14] Z. Kulpa, Diagrammatic representation of interval space in proving theorems about interval relations. *Reliable Computing*, **3**(3): 209-217, 1997.
[15] Z. Kulpa, Diagrammatic representation for a space of intervals. *Machine GRAPHICS & VISION*, **6**(1): 5-24, 1997.
[PostScript file: see http://www.ippt.gov.pl/~zkulpa/zkpubl.html#diagrammatic]
[16] Z. Kulpa, A. Radomski, O. Gajl, M. Kleiber, I. Skalna. Hybrydowy system ekspertowy jakoöciowo-iloöciowej analizy ukðadów mechanicznych [Hybrid expert system for qualitative/quantitative analysis of mechanical structures, in Polish], In: Z. Bubnicki, A. Grzech, eds. *Inüynieria wiedzy i systemy ekspertowe* [*Knowledge Engineering and Expert Systems*, in Polish], Vol. 2, 135-142. (Proc. IIIrd Natl. Scientific Conference, Wrocðaw, June 10-12, 1997), Oficyna Wydawnicza Politechniki Wrocðawskiej, Wrocðaw 1997.
[17] Z. Kulpa, A. Radomski, O. Gajl, M. Kleiber, I. Skalna, Hybrid expert system for qualitative and quantitative analysis of mechanical structures, In: L. Ironi, ed. *Qualitative Reasoning: 11th International Workshop* (Cortona, Italy, June 3-6, 1997), 287-293. Pub. N. 1036: Istituto di Analisi Numerica C.N.R., Pavia, Italy, 1997.
[PostScript file: see http://www.ippt.gov.pl/~zkulpa/zkpubl.html#quaphys]
[18] S.E. Laveuve. Definition einer Kahan-Arithmetik und ihre Implementierung. In: K. Nickel, ed., *Interval Mathematics 1975*. Lecture Notes in Computer Science, vol. **29**: 236-245. Springer Verlag, Berlin 1975.
[19] R.E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
[20] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge 1990.
[21] W. Oettli. On the solution set of linear system with inaccurate coefficients. *SIAM J. Numer. Anal.* Series 8, **2**(1): 115-118, 1965.
[22] W. Oettli, W. Prager. Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numer. Math.* **6**: 405-409, 1964.
[23] A. Pownuk. Przedziaðowe metody w rozwiäzywaniu ukðadów równañ liniowych rozmytych. [Interval methods in solving linear fuzzy systems of equations, in Polish]. Konferencja Naukowo-Dydaktyczna „Nowe Tendencje w Nauczaniu Mechaniki" (Koðobrzeg, 25-26.10.1996), 175-184. Wyd. INNOWEX, Koszalin 1996.
[24] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. (2nd ed.) Cambridge University Press, Cambridge 1993.
[25] S.S. Rao, L. Berke. Analysis of uncertain structural systems using interval analysis. *AIAA Journal*, **35**(4): 727-735, 1997.
[26] J. Rohn. Systems of linear interval equations. *Linear Algebra and its Applications*, **126**: 39-78, 1989.
[27] J. Rohn. Cheap and tight bounds: The recent result by E. Hansen can be made more efficient. *Interval Computations*, **4**: 13-21, 1993.
[28] J. Rohn. Linear interval equations: Computing enclosures with bounded relative overestimation is NP-hard. In: R.B. Kearfott, V. Kreinovich, eds., *Applications of Interval Computations*. Kluwer Academic Publishers, 81-89, 1996.
[29] J. Rohn. [Untitled]. A letter to the *Reliable-Computing* e-mail discussion list, August 1997.
[30] J. Rohn, V. Kreinovich. Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard. *SIAM J. Matr. Anal. Appl.*, **16**: 415-420, 1995.
[31] S.M. Rump. Verification methods for dense and sparse systems of equations. In: J. Herzberger, ed., *Topics in Validated Computations*. Elsevier Science B.V., 63-135, 1994.
[32] S. P. Shary. Algebraic approach to the interval linear static identification, tolerance, and control problems, or one more application of Kaucher arithmetic. *Reliable Computing*, **2**(1): 3-33, 1996.
[33] Yu. I. Shokin. On interval problems, interval algorithms and their computational complexity. In: G. Alefeld, A. Frommer and B. Lang, eds. *Scientific Computing and Validated Numerics*. Akademie-Verlag, Berlin 1996, pp. 314--328.
[34] J. Skrzypczyk. Fuzzy finite element methods – A new methodology. In: *Computer Methods in Mechanics* (Proc. XIII Polish Conference on Computer Methods in Mechanics, Poznañ, Poland, May 5-8, 1997), Vol. 4: 1187-1194. Poznañ University of Technology, Poznañ 1997.
[35] M. Warmus. Calculus of approximations. *Bull. Acad. Pol. Sci., Cl. III*, **4**(5): 253-259, 1956.