ELSEVIER

Inner and outer bounds for the solution set of parametric linear systems

Evgenija Popova[a,1], Walter Krämer[b,*]

[a]*Institute of Mathematics and Informatics, Acad. G. Bonchev str., bl. 8, BG-1113 Sofia, Bulgaria*
[b]*Faculty of Mathematics and Natural Sciences, University of Wuppertal, D-42119 Wuppertal, Germany*

Abstract

Consider linear systems involving affine-linear dependencies on interval parameters. Presented is a free C-XSC software implementing a generalized parametric fixed-point iteration method for verified enclosure of the parametric solution set. Some specific features of the corresponding algorithm concerning sharp enclosure of the contracting matrix and inner approximation of the solution enclosure are discussed.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Parametric linear systems; Validated software; C-XSC; Inner estimation

1. Introduction

Consider the linear system $A(p) \cdot x = b(p)$ represented as

$$\left(A^{(0)} + \sum_{v=1}^{k} p_v A^{(v)} \right) \cdot x = b^{(0)} + \sum_{v=1}^{k} p_v b^{(v)}, \quad p \in [p] \in \mathbb{IR}^k, \tag{1}$$

where $A^{(0)} := (a_{ij}^{(0)}), \ldots, A^{(k)} := (a_{ij}^{(k)}) \in \mathbb{R}^{n \times n}$ are numerical matrices, $b^{(0)} := (b_i^{(0)}), \ldots, b^{(k)} := (b_i^{(k)}) \in \mathbb{R}^n$ are numerical vectors. When the parameters $p_1, \ldots, p_k$ vary within a range $[p] \in \mathbb{IR}^k$ the parametric solution set is

$$\Sigma^p = \Sigma(A(p), b(p), [p]) := \{ x \in \mathbb{R}^n \mid A(p) \cdot x = b(p) \text{ for some } p \in [p] \}. \tag{2}$$

For a nonempty bounded $\Sigma \subseteq \mathbb{R}^n$, define interval hull $\Diamond : P\mathbb{R}^n \to \mathbb{IR}^n$ by $\Diamond(\Sigma) := [\inf \Sigma, \sup \Sigma]$. Here we discuss the computation of $[y] \in \mathbb{IR}^n$ such that $[y] \supseteq \Diamond(\Sigma^p) \supseteq \Sigma^p$.

An iteration method for verified enclosure of Σ^p, which accounts for arbitrary affine-linear dependencies in the matrix and the right-hand side vector, is proposed by Rump in [12] and generalized for strongly regular parametric matrices in [9].

---

* Corresponding author.

E-mail addresses: epopova@bio.bas.bg (E. Popova), kraemer@math.uni-wuppertal.de (W. Krämer).

The goal of this work is to provide a free, open-source software for the verified enclosure of the parametric solution set in the environment of C-XSC [4]. The software tool, we describe here, implements a generalized fixed-point iteration method for parametric linear systems and has our expertise and experience in implementing validated interval computations built in [10].

We use the following notations. $\mathbb{R}^n$, $\mathbb{R}^{n \times m}$ denote the set of real vectors with n components and the set of real $n \times m$ matrices, respectively. By normal (*proper*) interval we mean a real compact interval $[a] = [a^-, a^+] := \{a \in \mathbb{R} \mid a^- \leqslant a \leqslant a^+\}$. By $\mathbb{IR}^n$, $\mathbb{IR}^{n \times m}$ we denote interval n-vectors and interval $n \times m$ matrices. The end-point functionals $(\cdot)^-$, $(\cdot)^+$, the mid-point function mid$(\cdot)$, where mid$([a^-, a^+]) := (a^- + a^+)/2$, and the diameter function diam$(\cdot)$, where diam$([a^-, a^+]) := a^+ - a^-$, are applied to interval vectors and matrices componentwise. $\varrho(A)$ is the spectral radius of a matrix A. Denote by $A([p]) := \Diamond\{A(p) \in \mathbb{R}^{n \times n} \mid p \in [p]\}$, $b([p]) := \Diamond\{b(p) \in \mathbb{R}^n \mid p \in [p]\}$ the non-parametric interval matrix, resp. vector, that corresponds to the parametric ones. Hence, $A([p]) \cdot x = b([p])$ is the non-parametric system corresponding to the parametric one.

2. Theoretical background

A detailed presentation of the theory of the enclosure methods for our problem can be found in [12,8,9,10]. Rump's parametric fixed-point method for enclosing the solution of (1) requires strong regularity of the non-parametric interval matrix [12]. The original theorem is generalized for strongly regular parametric matrices [9] by replacing $[C] = I - R \cdot A([p])$ with (3). Although similar iteration methods are used also by other authors [1,7] without addressing the rounding errors and not referring to [12], the necessity of using an iteration matrix of type (3) is not justified therein. The latter is proven in [8], where classes of matrices are defined for which the generalization is efficient. The advantages of the generalized method are demonstrated in Section 4, [9,10] by examples comparing both methods.

Theorem 1. *Consider* (1). *Let* $R \in \mathbb{R}^{n \times n}$, $[Y] \in \mathbb{IR}^n$, $\tilde{x} \in \mathbb{R}^n$ *and define*

$$\mathbb{IR}^n \ni [Z] := R \cdot (b^{(0)} - A^{(0)}\tilde{x}) + \sum_{v=1}^{k} [p_v](R \cdot b^{(v)} - R \cdot A^{(v)} \cdot \tilde{x}),$$

$$\mathbb{IR}^{n \times n} \ni [C(p)] := I - R \cdot A^{(0)} - \sum_{v=1}^{k} [p_v](R \cdot A^{(v)}). \tag{3}$$

Initialize $[U] := [Y]$ *and define* $[V] \in \mathbb{IR}^n$ *by means of the following Einzelschrittverfahren*:

$$1 \leqslant i \leqslant n : V_i := \{[Z] + [C(p)] \cdot [U]\}_i, \quad U_i := V_i \ (update \ of \ [U]).$$

If $[V] \subsetneq [Y]$, *then* R *and every matrix* $A(p)$, $p \in [p]$ *are regular, and for every* $p \in [p]$ *the unique solution* $\hat{x} = A^{-1}(p)b(p)$ *of* (1) *satisfies* $\hat{x} \in \tilde{x} + [V]$.

With $[D] := [C(p)] \cdot [V] \in \mathbb{IR}^n$ *the following inner estimation of* $\Diamond(\Sigma^p)$ *holds true*:

$$[\tilde{x} + [Z]^- + [D]^+, \tilde{x} + [Z]^+ + [D]^-] \subseteq [\inf(\Sigma^p), \sup(\Sigma^p)].$$

In the implementation we choose $R \approx A^{-1}(p_m)$ and $\tilde{x} \approx A^{-1}(p_m) \cdot b(p_m)$, where $p_m = \text{mid}([p])$. To force $[V] \subsetneq [Y]$, the concept of ε-*inflation* is introduced. For a real interval $[w]$, ε-*inflation* is defined by

$$\text{blow}([w], \varepsilon) = \begin{cases} [w] + \text{diam}([w])[-\varepsilon, \varepsilon] & \text{if diam}([w]) > 0, \\ [\text{pred}(w), \text{succ}(w)] & \text{if diam}([w]) = 0, \end{cases}$$

where pred(w), succ(w) are the predecessor and successor of a floating-point number w in the floating-point screen, see [10].

3. Inner and outer estimations in floating point

The first part of Theorem 1 defines how to find an interval vector $[y]$, which is verified to contain $\Diamond(\Sigma^p)$. However, it is important to know how much this inclusion overestimates the exact hull of the parametric solution set. The quality of the outer enclosure can be estimated by a componentwise inner estimation of the exact hull [11]. $[x] \in \mathbb{IR}^n$ is called componentwise inner approximation for some set $\Sigma \in \mathbb{R}^n$ if

$$\inf_{\sigma \in \Sigma} \sigma_i \leqslant x_i^- \quad \text{and} \quad x_i^+ \leqslant \sup_{\sigma \in \Sigma} \sigma_i \quad \text{for every } 1 \leqslant i \leqslant n.$$

It should be noted that $[x] \subseteq [\inf(\Sigma), \sup(\Sigma)]$ but $[x] \not\subseteq \Sigma$.

Let $\mathbb{F} \subset \mathbb{R}$ denote the set of floating-point numbers on a computer. Denote by $\bigtriangledown, \bigtriangleup : \mathbb{R} \longrightarrow \mathbb{F}$ the directed roundings toward $-\infty$, resp. $+\infty$ [6]. For intervals $[a] = [a^-, a^+] \in \mathbb{IR}$, outward ($\Diamond$) and inward ($\bigcirc$) roundings $\Diamond, \bigcirc : \mathbb{IR} \longrightarrow \mathbb{IF}$ are defined as

$$\Diamond([a]) := [\bigtriangledown(a^-), \bigtriangleup(a^+)] \supseteq [a], \quad \bigcirc[a] := [\bigtriangleup(a^-), \bigtriangledown(a^+)] \subseteq [a]. \tag{4}$$

If $\circ \in \{+, -, \times, /\}$ is an arithmetic operation and $[a], [b] \in \mathbb{IF}$, the corresponding computer operations $\Diamond\!\!\!\Diamond, \odot : \mathbb{IF} \times \mathbb{IF} \longrightarrow \mathbb{IF}$ are defined by

$$[a]\Diamond\!\!\!\Diamond[b] := \Diamond([a] \circ [b]) = [\bigtriangledown(([a] \circ [b])^-), \quad \bigtriangleup(([a] \circ [b])^+)] \supseteq [a] \circ [b], \tag{5}$$

$$[a]\odot[b] := \bigcirc([a] \circ [b]) = [\bigtriangleup(([a] \circ [b])^-), \quad \bigtriangledown(([a] \circ [b])^+)] \subseteq [a] \circ [b]. \tag{6}$$

To compute an inner estimation of the solution enclosure by the second part of Theorem 1 on a computer one needs an inner estimation $\bigcirc[Z]$ of $[Z]$. Obtaining guaranteed inner approximations on a computer in conventional interval arithmetic is possible only if the four interval operations are implemented with inward rounding $\odot$ in addition to the four $\Diamond\!\!\!\Diamond$ operations. Since most of the wide-spread interval packages do not support inwardly rounded interval arithmetic, here we give an alternative computational technique based on the properties of an algebraic extension of the conventional interval arithmetic.

The set of proper intervals $\mathbb{IR}$ is extended in [5] by the set $\{[a^-, a^+] \mid a^-, a^+ \in \mathbb{R}, a^- \geqslant a^+\}$ of *improper* intervals obtaining thus the set $\mathbb{IR}^* = \{[a^-, a^+] \mid a^-, a^+ \in \mathbb{R}\}$ of all ordered couples of real numbers called here generalized intervals. The conventional (arithmetic and lattice) operations, order relations and other functions are isomorphically extended onto the whole set of proper and improper intervals [5]. The same is done for the inward and outward roundings so that formulae (4)–(6) are valid for generalized intervals. We present only those basic facts from generalized interval arithmetic which are necessary to use it as an intermediate computational tool for handling proper interval problems.

"*Dual*" is an important monadic operator that reverses the end-points of the intervals and expresses an element-to-element symmetry between proper and improper intervals in $\mathbb{IR}^*$. For $[a] = [a^-, a^+] \in \mathbb{IR}^*$, its dual is defined by $Dual([a]) = [a^+, a^-]$. *Dual* is applied componentwise to vectors and matrices. For $[a], [b] \in \mathbb{IR}^*$ and $\circ \in \{+, -, \times, /\}$,

$$Dual(Dual([a])) = [a], \quad Dual([a] \circ [b]) = Dual([a]) \circ Dual([b]). \tag{7}$$

As the following properties show inner numerical approximations can be obtained at no additional cost only by outward directed rounding and the *Dual* operator in $\mathbb{IF}^*$ [2].

$$\text{For } [a] \in \mathbb{IR}^*, \quad \bigcirc([a]) = Dual(\Diamond(Dual([a]))). \tag{8}$$

$$\text{For } [a], [b] \in \mathbb{IF}^*, \ \circ \in \{+, -, \times, /\}, [a]\odot[b] = Dual(Dual([a])\Diamond\!\!\!\Diamond Dual([b])). \tag{9}$$

We apply the above properties to obtain inner estimations of proper interval problems in a computing environment not supporting generalized interval arithmetic.

Now, we consider the computation of $\bigcirc([Z])$. With the notations of Theorem 1, let

$$z^{(v)} := R(b^{(v)} - A^{(v)}\tilde{x}), \quad v = 0, 1, \dots, k.$$

Table 1
Multiplication *Dual* $([a^-, a^+]) \times [b^-, b^+]$ for $[a], [b] \in \mathbb{IR}$

$Dual([a]) \times [b]$	$b^- \geqslant 0$	$b^+ \leqslant 0$	$b^- < 0 < b^+$
$a^- \geqslant 0$	$[a^+ b^-, a^- b^+]$	$[a^- b^-, a^+ b^+]$	$[a^- b^-, a^- b^+]$
$a^+ \leqslant 0$	$[a^+ b^+, a^- b^-]$	$[a^- b^+, a^+ b^-]$	$[a^+ b^+, a^+ b^-]$
$a^- < 0 < a^+$	$[a^+ b^-, a^- b^-]$	$[a^- b^+, a^+ b^+]$	$[0, 0]$

Then $[Z] = z^{(0)} + \sum_{v=1}^{k} [p_v] z^{(v)}$. Using the well-known inclusion properties of interval arithmetic we obtain

$$[Z] = z^{(0)} + \sum_{v=1}^{k} [p_v] z^{(v)} \subseteq \Diamond \left(\Diamond(z^{(0)}) + \sum_{v=1}^{k} [p_v] \cdot \Diamond(z^{(v)}) \right) \quad \text{and} \quad \bigcirc \left(\bigcirc(z^{(0)}) + \sum_{v=1}^{k} [p_v] \times \bigcirc(z^{(v)}) \right)$$
$$\subseteq [Z].$$

The operator $\times$ indicates multiplication of proper and improper intervals. We will see that we must implement this operation only for the case improper interval multiplied by a proper interval. The definition of this operation is given in Table 1.

Applying properties (8) to the last inclusion in $\mathbb{IR}^{*n}$, we get

$$Dual \left(\Diamond \left(Dual(\bigcirc(z^{(0)})) + \sum_{v=1}^{k} Dual([p_v]) \times Dual(\bigcirc(z^{(v)})) \right) \right) \subseteq [Z]. \tag{10}$$

For fixed $v = 0, \ldots, k$, let $d^{(v)} \in \mathbb{F}^n$ be a floating-point approximation of $b^{(v)} - A^{(v)} \tilde{x}$, $d^{(v)} \approx b^{(v)} - A^{(v)} \tilde{x}$. The error $e^{(v)}$ of this approximation is $e^{(v)} = b^{(v)} - A^{(v)} \tilde{x} - d^{(v)}$.

Hence, $\bigcirc(e^{(v)}) := \bigcirc(b^{(v)} - A^{(v)} \tilde{x} - d^{(v)})$ is an inner approximation of the error, while $\Diamond(e^{(v)}) := \Diamond(b^{(v)} - A^{(v)} \tilde{x} - d^{(v)})$ is an outer one

$$\bigcirc(e^{(v)}) \subseteq b^{(v)} - A^{(v)} \tilde{x} - d^{(v)} \subseteq \Diamond(e^{(v)}).$$

Multiplying both sides above by R and applying the inclusion properties of interval operations, we get

$$\bigcirc(z^{(v)}) := \bigcirc(R \cdot \bigcirc(e^{(v)}) + R \cdot d^{(v)}) \subseteq z^{(v)} \subseteq \Diamond(R \cdot \Diamond(e^{(v)}) + R \cdot d^{(v)}) =: \Diamond(z^{(v)}). \tag{11}$$

If computed by a real dotproduct accumulator, $\bigcirc(e^{(v)}) = Dual(\Diamond(e^{(v)}))$ and thus left-hand side inclusion in (11) is equivalent to

$$\bigcirc(R \cdot Dual(\Diamond(e^{(v)})) + R \cdot d^{(v)}) \subseteq z^{(v)}.$$

Applying (8) we get

$$Dual(\Diamond(R \cdot \Diamond(e^{(v)}) + R \cdot d^{(v)})) \subseteq z^{(v)},$$

that is,

$$\bigcirc(z^{(v)}) = Dual(\Diamond(R \cdot \Diamond(e^{(v)}) + R \cdot d^{(v)})) = Dual(\Diamond(z^{(v)})).$$

Substituting the last expression into (10) and applying (7), we obtain $\bigcirc([Z])$ only by outwardly rounded interval operations between proper and improper intervals.

$$\bigcirc([Z]) = Dual \left(\Diamond \left(\Diamond(z^{(0)}) + \sum_{v=1}^{k} Dual([p_v]) \times \Diamond(z^{(v)}) \right) \right).$$

The product $Dual([p_v]) \times \Diamond(z^{(v)})$ for $v = 0, 1, \ldots, k$, where $\Diamond(z^{(v)})$ is a proper interval vector and $Dual([p_v])$ is an improper interval, should be implemented according to Table 1 defining the product of a proper and an improper interval as a special case of the multiplication of generalized intervals considered in [5].

Above we have proven the following:

Theorem 2. *Consider* (1) *with* $p \in [p] \in \mathbb{IF}^k$. *Let* $R \in \mathbb{F}^{n \times n}$, $[Y] \in \mathbb{IF}^n$ *and* $\tilde{x} \in \mathbb{F}^n$. *For* $v = 0, 1, \ldots, k$, *define* $d^{(v)} :\approx b^{(v)} - A^{(v)}\tilde{x}$, $\Diamond(e^{(v)}) := \Diamond(b^{(v)} - A^{(v)}\tilde{x} - d^{(v)})$ *and* $\Diamond(z^{(v)}) := \Diamond(R \cdot \Diamond(e^{(v)}) + R \cdot d^{(v)})$. *Define* $\Diamond([Z]) \in \mathbb{IF}^n$, $[C(p)] \in \mathbb{IF}^{n \times n}$ *by*

$$\Diamond([Z]) := \Diamond\left(\Diamond(z^{(0)}) + \sum_{v=1}^{k}[p_v] \cdot \Diamond(z^{(v)})\right),$$

$$[C(p)] := \Diamond\left(I - R \cdot A^{(0)} - \sum_{v=1}^{k}[p_v] \times (R \cdot A^{(v)})\right).$$

Define $[V] \in \mathbb{IF}^n$ *by* $1 \leqslant i \leqslant n$: $V_i := \{\Diamond([Z]) \diamondplus [C(p)] \diamondtimes [U]\}_i$, $[U] := (V_1, \ldots, V_{i-1}, Y_i, \ldots, Y_n)^\top$. *If* $[V] \subsetneqq [Y]$, *then* R *and every matrix* $A(p)$, $p \in [p]$ *are regular, and the solution set* (2) *satisfies* $[\inf(\Sigma^p), \sup(\Sigma^p)] \subseteq \tilde{x} \diamondplus [V]$.

With $[D] := [C(p)] \diamondtimes [V]$ *the following inner estimation holds true*:

$$Dual\left(\tilde{x} \diamondplus \Diamond\left(\Diamond(z^{(0)}) + \sum_{v=1}^{k} Dual([p_v]) \times \Diamond(z^{(v)})\right) \diamondplus [D]\right)$$

$$\subseteq [\inf(\Sigma^p), \sup(\Sigma^p)], \tag{12}$$

where the operator $\times$ *is defined in Table* 1.

The inner estimation (12) should be interpreted in terms of proper intervals. Components of the inner estimations (12) may be improper intervals which are interpreted in $\mathbb{IR}$ as empty sets. In this case no inner estimation for these components can be given.

4. New open source software ParLinSys

ParLinSys is a new open source module for verified solving of parametric linear systems which is implemented in C++ using C-XSC [4,6] and some routines from the C++ Toolbox for Verified Computing [3]. The algorithm, based on the above theoretical considerations, and the implementation details are given in [10]. ParLinSys uses entirely numerical representation for the parametric matrix and r.h. side vector. Since our implementation is intended also for education and experimentation purposes, the implemented function for computing enclosures for the parametric solution set involves arguments for switching on/off the sharp enclosure of the iteration matrix, specifying a value for the constant of epsilon inflation, and switching on/off the computation of an inner approximation for the outer enclosure. For a more detailed documentation and examples about how to use the new module refer to [10]. The software and the paper [10] are freely available at http://www.math.uni-wuppertal.de/wrswt/xsc/cxsc_software.html.

Below we give three numerical examples demonstrating the advantage of the generalized iteration method. The matrices specified by the following parameter dependent system are all symmetric and regular:

$$\begin{pmatrix} 1 & p \\ p & \frac{31}{10}p + \frac{9}{100} \end{pmatrix} \cdot x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad [p] = \left[\frac{29}{10}, \frac{31}{10}\right] \in \mathbb{IR}^1.$$

The corresponding non-parametric interval matrix

$$A([p]) = \begin{pmatrix} 1 & \left[\frac{29}{10}, \frac{31}{10}\right] \\ \left[\frac{29}{10}, \frac{31}{10}\right] & \left[\frac{908}{100}, \frac{970}{100}\right] \end{pmatrix}$$

is not strongly regular. The spectral radius $\varrho(|I - \text{mid}^{-1}(A[p]) \cdot A([p])|) = \frac{7}{6} + \frac{\sqrt{8437}}{78}$ is greater than 1 (as it is readily seen the matrix $A([p])$ even contains singular point matrices). This means that all methods based on strong

Table 2
ParLinSys results for the three-diagonal parametric problem

| n | 0.1% | | | | 1% | | | | 10% | | | |
| | [C] | | [C(p)] | | [C] | | [C(p)] | | [C] | | [C(p)] | |
	it	min–max	it	min–max	it	min–max	it	min–max	it	min–max	it	min–max
5	1	7.99–9.97	1	8.64–9.97	2	0–9.70	2	1.61–9.72	3	0–6.19	2	0–7.20
20	1	9.67–9.98	1	9.97–9.98	1	6.86–9.76	1	9.67–9.78	4	0–7.12	1	6.95–7.91
30	1	9.49–9.97	1	9.97–9.98	2	5.31–9.65	1	9.67–9.78	5	0–0	1	6.93–7.88
35	2	3.57–9.80	1	9.03–9.96	2	0–7.71	2	3.19–9.60		—	3	0–6.00
40	1	9.33–9.97	1	9.97–9.98	2	3.99–9.64	1	9.67–9.78		—	1	6.93–7.88
50	1	9.15–9.92	1	9.97–9.98	2	2.61–9.11	1	9.67–9.77		—	1	6.89–7.80
100	1	8.21–9.79	1	9.96–9.97	3	0–6.41	1	9.63–9.75		—	2	6.39–7.45
150	1	6.92–8.58	1	9.96–9.96		—	1	9.56–9.63		—	2	5.63–6.27
250	3	0–0.43	1	9.89–9.90		—	1	8.96–8.97		—	3	0–0
450	3	0–1.10	1	9.82–9.92		—	1	8.32–9.19		—	3	0–1.57
650	2	0–4.38	2	2.84–9.96		—	2	0–9.62		—	3	0–5.01
800	5	0–0	2	0–9.87		—	2	0–8.74		—	4	0–0
830		—	2	0–9.84		—	2	0–8.37		—	6	0–0
850		—	1	4.04–9.80		—	2	0–7.98		—		—
900		—	2	0.81–9.54		—	2	0–5.33		—		—
1500		—	2	0–9.55		—	2	0–5.43		—		—
2000		—	2	0–9.83		—	2	0–8.23		—		—

regularity of the matrix $A([p])$ will fail. However, the improved iteration matrix $[C(p)]$ leads to a spectral radius $\varrho(|[C(p)]|) = \frac{31}{78} + \frac{\sqrt{877}}{78} = 0.777\ldots < 1$. Running our solver ParLinSys we find e.g. the following outer estimation:

```
([-4.188067739E+01,7.464990816E+01], [-2.464361101E+01,1.438720075E+01]).
```

Consider $A(p) \cdot x = b(p,q)$ with

$$A(p) = \begin{pmatrix} 1 & p & & & \\ p & 1 & p & & \\ & p & 1 & p & \\ & & & \ddots & \\ & & & p & 1 \end{pmatrix}, \quad b(p,q) = (-p, 0, \ldots, 0, -q)^{\top},$$

where $p \in [100 \pm \delta], q \in [1 \pm \delta/100]$. ParLinSys module was used to solve the above parametric problem for dimensions up to 2000, three different tolerances $\delta \in \{0.1, 1, 10\}$ for the parameters and applying either non-parametric iteration matrix $[C]$ or the improved matrix $[C(p)]$. A function Sharpness([x], [y]) was involved to compute a measure for the quality of a solution enclosure $[x] \supseteq \Diamond(\Sigma)$ based on an inner estimation $[y] \subseteq \Diamond(\Sigma)$.

Sharpness $([x], [y]) := \{1$ if diam $([x]) = 0$, 0 if $[y] = \emptyset$, diam $([y])$/diam $([x])$ otherwise$\}$.

Table 2 presents the obtained results, where "it" denotes the number of iterations, and "min/max" are the corresponding min/max of sharpness estimations for the solution components. Sharpness values are multiplied by 10 and dash (–) indicates that the corresponding method has failed to find a solution enclosure.

Consider the parametric systems $Q(2, p)x = b(p)$, where for $i, j = 1, \ldots, n$

$$q_{ij}(2, p) := \begin{cases} p_j & \text{if } i \leqq j, \\ 0 & \text{if } i = j + 2, \\ 1 & \text{otherwise,} \end{cases} \quad \begin{aligned} & b(p) = (p_1, \ldots, p_n)^{\top}, \\ & p_k \in [k \pm k * \delta/100], \quad k = 1, \ldots, n, \\ & \delta \in \{2.5, 5, 10\}. \end{aligned}$$

The results for these dense systems with many parameters are presented in Table 3.

Table 3
ParLinSys results for the systems $Q(2, p)x = b(p)$

n	2.5%				5%				10%			
	[C(p)]		[C]		[C(p)]		[C]		[C(p)]		[C]	
	it	min–max	it	min–max	it	min–max	it	min–max	it	min–max	it	min–max
4	1	8.22–8.93	3	0–2.14	2	6.48–7.82	6	0–0	2	3.33–5.71		—
10	1	6.86–9.15	5	0–2.36	2	4.17–8.26		—	2	0.06–6.51		—
50	2	3.98–9.42		—	2	0.39–8.84		—	2	0–7.68		—
100	2	2.08–9.43		—	2	0–8.87		—	2	0–7.75		—
200	2	0–9.44		—	2	0–8.89		—	2	0–7.78		—
300	2	0–9.45		—	2	0–8.89		—	2	0–7.79		—
400	2	0–9.45		—	2	0–8.89		—	2	0–7.80		—
500	2	0–9.45		—	2	0–8.90		—	2	0–7.80		—

5. Conclusion

ParLinSys seems to be the first open source software able to compute inner and outer estimations for the solution set hull of parametric linear systems with affine-linear dependencies. Our generalized method requires strong regularity of the parametric matrix. In contrast to former methods the improved iteration matrix extends the scope of applicability to systems where the corresponding non-parametric interval matrix is not necessarily strongly regular. The examples show that the improved approach often allows bigger dimensions and larger intervals for the parameters (Tables 2 and 3). In many cases less iterations are necessary and in general the sharpness of the inner and outer estimations is better.

References

[1] O. Dessombz, et al., Analysis of mechanical systems using interval computations applied to finite element methods, J. Sound Vibration 239 (5) (2001) 949–968.
[2] E. Gardeñes, A. Trepat, Fundamentals of SIGLA, an interval computing system over the completed set of intervals, Computing 24 (1980) 161–179.
[3] R. Hammer, M. Hocks, U. Kulisch, D. Ratz, C + + Toolbox for Verified Computing: Basic Numerical Problems, Springer, Berlin, Heidelberg, New York, 1995.
[4] W. Hofschuster, W. Krämer, C-XSC 2.0 - A C + + class library for extended scientific computing, in: R. Alt, et al., (Eds.), Numerical Software with Result Verification, Lecture Notes in Computer Science, vol. 2991, Springer, Berlin, 2004, pp. 15–35.
[5] E. Kaucher, Interval analysis in the extended interval space IR, Computing 2 (Suppl.) (1980) 33–49.
[6] R. Klatte, U. Kulisch, C. Lawo, M. Rauch, A. Wiethoff, C-XSC, A C + + Class Library for Extended Scientific Computing, Springer, Berlin, 1993.
[7] R. Muhanna, et al., Penalty-based solution for the interval finite element, presentation at First Scandinavian Workshop on Interval Methods and their Applications, TU Denmark, 2003 ⟨http://www.imm.dtu.dk/∼km/int-03/Slides/⟩.
[8] E.D. Popova, Strong regularity of parametric interval matrices, in: Mathematics and Education in Mathematics, Proceedings of the 33rd Spring Conference of UBM, Borovetz, 2004, 446–451 ⟨http://www.math.bas.bg/∼epopova/papers/⟩.
[9] E.D. Popova, Generalizing the parametric fixed-point iteration, PAMM—Proc. Appl. Math. Mech. 4 (2004) 680–681.
[10] E.D. Popova, W. Krämer, Parametric fixed-point iteration implemented in C-XSC, preprint 2003/03, WRSWT, Universität Wuppertal, 2003, 1–27.
[11] S. Rump, Rigorous sensitivity analysis for systems of linear and nonlinear equations, Math. Comput. 54 (190) (1990) 721–736.
[12] S. Rump, Verification methods for dense and sparse systems of equations, in: J. Herzberger (Ed.), Topics in Validated Computations, North-Holland, Amsterdam, 1994, pp. 63–135.