Optimal Preconditioning for the Interval Parametric Gauss–Seidel Method

Milan Hladík(✉)

Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic
hladik@kam.mff.cuni.cz

Abstract. We deal with an interval parametric system of linear equations, and focus on the problem how to find an optimal preconditioning matrix for the interval parametric Gauss–Seidel method. The optimality criteria considered are to minimize the width of the resulting enclosure, to minimize its upper end-point or to maximize its lower end-point. We show that such optimal preconditioners can be computed by solving suitable linear programming problems. We also show by examples that, in some cases, such optimal preconditioners are able to significantly decrease an overestimation of the results of common methods.

Keywords: Interval computation · Interval parametric system · Preconditioner · Linear programming

1 Introduction

Consider an interval linear system of equations

$$Ax = b, \quad A \in \boldsymbol{A},\ b \in \boldsymbol{b}, \tag{1}$$

where

$$\begin{aligned} \boldsymbol{A} &:= [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{n\times n};\ \underline{A} \le A \le \overline{A}\},\\ \boldsymbol{b} &:= [\underline{b}, \overline{b}] = \{b \in \mathbb{R}^{n};\ \underline{b} \le b \le \overline{b}\} \end{aligned}$$

are an interval matrix and an interval vector, respectively. The solution set is defined as

$$\Sigma := \{x \in \mathbb{R}^n;\ \exists A \in \boldsymbol{A},\ \exists b \in \boldsymbol{b} : Ax = b\}.$$

Since it is nonconvex in general, the problem is usually to compute an interval vector enclosing the solution set. Computing the smallest enclosure is an NP-hard problem [1], so the known polynomial-time methods overestimate more-or-less the optimal enclosure. There are, however, plenty of methods varying in time complexity and tightness of the resulting enclosures [1,4,11].

M. Nehmeier et al. (Eds.): SCAN 2014, LNCS 9553, pp. 116–125, 2016.
DOI: 10.1007/978-3-319-31769-4_10

Notation. The midpoint and the radius matrices corresponding to an interval matrix $\boldsymbol{A}$ are defined respectively as

$$A^c := \frac{1}{2}(\underline{A} + \overline{A}), \quad A^\Delta := \frac{1}{2}(\overline{A} - \underline{A}).$$

Similarly we define interval vectors and intervals. The ith column of a matrix $C \in \mathbb{R}^{n \times n}$ is denoted by C_{*i}.

Preconditioning. Many methods for enclosing the solution set use preconditioning. Let $C \in \mathbb{R}^{n \times n}$. Then the interval system (1) preconditioned by C reads

$$A'x = b', \quad A' \in (C\boldsymbol{A}),\ b' \in (C\boldsymbol{b}),$$

where $C\boldsymbol{A}$ and $C\boldsymbol{b}$ are calculated by interval arithmetic [11]. The solution set corresponding to the preconditioned system contains the original one as a subset, so by preconditioning we do not miss any solution. Even though the solution set inflates by preconditioning, most of the methods used perform better when the system is preconditioned by a suitable matrix.

It is commonly recommended to use the preconditioner $C = (A^c)^{-1}$ or its numerical approximation. Some theoretical properties justifying this choice were stated by Neumaier [10,11]. This does not mean, however, that the midpoint inverse preconditioner yields the best results for each method and for each input data.

Kearfott [5] initiated a research in constructing an optimal preconditioning matrix [6–8]. The authors investigated the interval Gauss–Seidel method with an application in nonlinear equation solving by the interval Newton method. They showed that the optimal precodnitioner for the interval Gauss–Seidel method can be formulated in terms of a linear programming, so it is polynomially computable. A hybrid preconditioning strategy combining the midpoint inverse and a certain kind of optimal preconditioners was proposed by Gau and Stadtherr [2], and some numerical tests and an application in global optimization were presented by Lin and Stadtherr [9].

The Interval Gauss–Seidel Method. Let us recall the interval Gauss–Seidel method briefly. Let $\boldsymbol{x} \supseteq \Sigma$ be an initial enclosure of the solution set. One interval Gauss–Seidel iteration for the preconditioned system is based on the operations

$$\begin{aligned} \boldsymbol{z}_i &:= \frac{1}{(C\boldsymbol{A})_{ii}} \left((C\boldsymbol{b})_i - \sum_{j \neq i} (C\boldsymbol{A})_{ij} \boldsymbol{x}_j \right), \\ \boldsymbol{x}_i &:= \boldsymbol{x}_i \cap \boldsymbol{z}_i, \end{aligned}$$

for $i = 1, \ldots, n$.

Interval Parametric Systems. An interval linear parametric system of equations is a family of systems

$$A(p)x = b(p), \quad p \in \boldsymbol{p},$$

where the constraint matrix and the right-hand side vector linearly depend on parameters $p_1, \ldots, p_K$,

$$A(p) = \sum_{k=1}^{K} A^k p_k, \quad b(p) = \sum_{k=1}^{K} b^k p_k.$$

Herein, $A^1, \ldots, A^K \in \mathbb{R}^{n \times n}$ are given matrices, $b^1, \ldots, b^K \in \mathbb{R}^n$ are given vectors, and $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K)$ is a given interval vector. The corresponding solution set is defined as

$$\Sigma_p := \{x \in \mathbb{R}^n;\ \exists p \in \boldsymbol{p} : A(p)x = b(p)\}.$$

Methods for computing an enclosure of the solution set were discussed, e.g., in [3, 17]. A parametrized version of interval Gauss–Seidel iteration in particular was addressed in Popova [13]. For parametric systems, preconditioning is applied, too.

In principle, a parametric system can be relaxed and the problem reduced to solving the standard interval system

$$Ax = b, \quad A \in \boldsymbol{A},\ b \in \boldsymbol{b},$$

where

$$\boldsymbol{A} := \sum_{k=1}^{K} A^k \boldsymbol{p}_k, \quad \boldsymbol{b} := \sum_{k=1}^{K} b^k \boldsymbol{p}_k$$

are evaluated by interval arithmetic. For a preconditioned system by $C \in \mathbb{R}^{n \times n}$, the tightest relaxation is done by evaluating

$$\boldsymbol{A} := \sum_{k=1}^{K} (CA^k) \boldsymbol{p}_k, \quad \boldsymbol{b} := \sum_{k=1}^{K} (Cb^k) \boldsymbol{p}_k.$$

Notice that a relaxation leads to overestimation of the solution set in general since we lose information about dependencies between the interval parameters.

The interval Gauss–Seidel iteration for preconditioned parametric system reads

$$\boldsymbol{z}_i := \frac{1}{\left(\sum_{k=1}^{K} (CA^k)_{ii} \boldsymbol{p}_k\right)} \left(\sum_{k=1}^{K} (Cb^k)_i \boldsymbol{p}_k - \sum_{j \neq i} \left(\sum_{k=1}^{K} (CA^k)_{ij} \boldsymbol{p}_k \right) \boldsymbol{x}_j \right), \tag{2}$$

$$\boldsymbol{x}_i := \boldsymbol{x}_i \cap \boldsymbol{z}_i,$$

for $i = 1, \ldots, n$.

For parametric systems, a residual form of enclosures is often employed. Let $x^0 \in \mathbb{R}^n$, for example the solution of $A(p^c)x = b(p^c)$. Then the residual form enclosure of Σ_p has the form of $\boldsymbol{x} = x^0 + \boldsymbol{y}$, where $\boldsymbol{y}$ encloses the solution set to the parametric system

$$A(p)x = b(p) - A(p)x^0, \quad p \in \boldsymbol{p}.$$

The interval Gauss–Seidel iteration (2) for this system works in the same manner as for the original system, only the vectors b^k are replaced by $b^k - A^k x^0$, $k = 1, \dots, K$.

Goal. The purpose of this paper is to extend the above mentioned results to interval parametric systems of linear equations by designing an optimal preconditioner for the parametric interval Gauss–Seidel method.

2 Optimal Preconditioners

In this section, we show how to construct an optimal preconditioner for (2). We focus on the direct version only since for the residual form it works analogously.

Since the ith step of (2) depends only on the ith row of C, we will design C row by row. For this purpose, let $i \in \{1, \dots, n\}$ be fixed, and consider the ith row of C, denoted by c.

Optimality of the preconditioner can be viewed from diverse perspectives; see various criteria surveyed in Kearfott et al. [7,8]. We will be concerned with the following objectives

- minimize the resulting width, that is, the objective is $\min 2z_i^{\Delta}$,
- minimize the resulting upper bound, that is, the objective is $\min \overline{z}_i$,
- maximize the resulting lower bound, that is, the objective is $\max \underline{z}_i$.

If we apply both the second and the third preconditioners, we obtain the smallest interval as a result after the intersection. This observation relies on standard interval arithmetic. Provided we allow division by zero-containing intervals and utilize generalized arithmetic, then tighter results are possible; see S-preconditioners in [7,8].

In the following, we will discuss the first and the second criteria only since the third criterion is easily reduced to the second one.

2.1 Minimal Width

Now, we deal with the first mentioned criterion – to minimize $2z_i^{\Delta}$. Suppose that $0 \in \boldsymbol{x}$ and $0 \in \boldsymbol{z}_i$. This is the case, for instance, when we apply the residual form and $x^0 \in \Sigma_p$. However, the resulting preconditioner seems to perform well even if the assumption is not satisfied despite it needn't be optimal.

In order that $\boldsymbol{z}_i$ is bounded, we will assume that the denominator in (2) does not contain the zero. Moreover, we will normalize c such that the denominator has the form of $[1, r]$ for some $r \geq 1$. Then, from our assumptions it follows that the operation in (2) is simplified to

$$\sum_{k=1}^{K}(cb^k)\boldsymbol{p}_k - \sum_{j\neq i}\left(\sum_{k=1}^{K}(cA^k_{*j})\boldsymbol{p}_k\right)\boldsymbol{x}_j$$

Denote

$$\begin{aligned}
\beta_k &:= |cb^k|, \quad k = 1,\dots,K,\\
\alpha_{jk} &:= |cA^k_{*j}|, \quad j = 1,\dots,n,\ k = 1,\dots,K,\\
\eta_j &:= \overline{\left(\sum\nolimits_{k=1}^{K}(cA^k_{*j})\boldsymbol{p}_k\right)\boldsymbol{x}_j}, \quad j \neq i,\\
\psi_j &:= \underline{\left(\sum\nolimits_{k=1}^{K}(cA^k_{*j})\boldsymbol{p}_k\right)\boldsymbol{x}_j}, \quad j \neq i.
\end{aligned}$$

Then our objective function reads

$$\min \sum_{k=1}^{K} 2p^{\Delta}_k\beta_k + \sum_{j\neq i}(\eta_j - \psi_j). \tag{3}$$

Now, we set up the constraints. By the definition of β_k, we have

$$\beta_k \geq cb^k,\ \beta_k \geq -cb^k, \quad k = 1,\dots,K. \tag{4}$$

Since β_k is minimized in the objective function, at least one of the inequalities will hold as equation, whence $\beta_k = |cb^k|$. Similarly for α_{jk} we obtain

$$\alpha_{jk} \geq cA^k_{*j},\ \alpha_{jk} \geq -cA^k_{*j}, \quad j = 1,\dots,n,\ k = 1,\dots,K. \tag{5}$$

The condition that the denominator is has the form of $[1, r]$ is formulated as the equation

$$c\sum_{k=1}^{K}A^k_{*i}p^c_k - \sum_{k=1}^{K}p^{\Delta}_k\alpha_{ik} = 1. \tag{6}$$

Eventually, we reformulate conditions on η_j and ψ_j. Since $0 \in \boldsymbol{x}_j$, the upper end-point of the interval product in the definition of η_j is attained either by the product of their upper end-points or their lower end-points. Thus, we get

$$\eta_j \geq c\sum_{k=1}^{K}A^k_{*j}p^c_k\underline{x}_j - \sum_{k=1}^{K}p^{\Delta}_k\underline{x}_j\alpha_{jk}, \quad j \neq i, \tag{7}$$

$$\eta_j \geq c\sum_{k=1}^{K}A^k_{*j}p^c_k\overline{x}_j + \sum_{k=1}^{K}p^{\Delta}_k\overline{x}_j\alpha_{jk}, \quad j \neq i. \tag{8}$$

Similarly for ψ_j,

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}^k p_k^c \underline{x}_j + \sum_{k=1}^{K} p_k^{\Delta} \underline{x}_j \alpha_{jk}, \quad j \neq i, \tag{9}$$

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}^k p_k^c \overline{x}_j - \sum_{k=1}^{K} p_k^{\Delta} \overline{x}_j \alpha_{jk}, \quad j \neq i. \tag{10}$$

Since η_j is maximized and ψ_j is minimized in the objective function, at least one of the inequalities is fulfilled as equation. Analogous considerations hold for α_{jk}. Therefore, we gathered all the constraints to formulate the optimization problem.

Optimization Problem. The optimal preconditioner of the first type is found by solving the optimization problem (3) under the constraints (4)–(10). This as a linear programming problem with $Kn + K + 3n - 2$ unknowns c, β_k, α_{jk}, η_j, and ψ_j, and $2Kn + 2K + 4n - 3$ constraints.

Notice that for standard interval linear Eq. (1), our approach would require approximately n^3 variables as there is a quadratic number of parameters. This is more than the linear programming formulation from Kearfott [6,7] using only a linear number of variables. His method, however, cannot be directly extended to parametric systems.

Overall, to determine the optimal preconditioner C, we have to solve n linear programs, which is a polynomial time problem. Moreover, C needn't be calculated in a verified way since any matrix can serve as a preconditioner.

On the other hand, solving n linear programs requires some computational effort, so it would be inefficient to compute an optimal C in each iteration of the interval Gauss–Seidel method. It seems more suitable to call the standard version using midpoint inverse preconditioner (or any other method), and after that to tighten the resulting enclosure by running several iterations with an optimal C.

2.2 Minimal Upper Bound

Herein, the criterion is to minimize $\overline{z}_i$. Suppose first that $\overline{z}_i > 0$. Using definitions of c, β_k, α_{jk}, and ψ_j from the previous section, the objective is formulated as

$$\min c\sum_{k=1}^{K} b^k p_k^c + \sum_{k=1}^{K} p_k^{\Delta} \beta_k - \sum_{j\neq i} \psi_j.$$

The constraints (4)–(6), (9)–(10) are employed in this problem, too. In addition, we have to take into account the remaining two possibilities for which ψ_j can be attain, and hence we involve also the inequalities

$$\psi_j \le c\sum_{k=1}^{K} A^k_{*j} p^c_k \underline{x}_j - \sum_{k=1}^{K} p^{\Delta}_k \underline{x}_j \alpha_{jk}, \quad j \ne i,$$
$$\psi_j \le c\sum_{k=1}^{K} A^k_{*j} p^c_k \overline{x}_j + \sum_{k=1}^{K} p^{\Delta}_k \overline{x}_j \alpha_{jk}, \quad j \ne i.$$

If $\overline{z}_i \le 0$, then we just replace (6) by the equation

$$c\sum_{k=1}^{K} A^k_{*i} p^c_k + \sum_{k=1}^{K} p^{\Delta}_k \alpha_{ik} = 1, \tag{11}$$

which normalizes the denominator in (2) to have the form of $[r, 1]$ for some $r \in (0, 1]$. In this case, we have to include the condition $r \ge 0$, which draws

$$c\sum_{k=1}^{K} A^k_{*i} p^c_k - \sum_{k=1}^{K} p^{\Delta}_k \alpha_{ik} \ge 0.$$

The situation $r = 0$ makes practically no harm (even theoretically by realizing what will be the result if extended arithmetic is used).

The weak point is that we do not know a priori whether $\overline{z}_i > 0$ or not. We recommend to use the condition $\overline{x}_i > 0$ instead. It means, if $\overline{x}_i > 0$, then we use (6), otherwise we use (11). The only possible fail may occur when $\overline{z}_i \le 0$ and $\overline{x}_i > 0$. In this case, the optimization problem does not find the optimal solution, however, the optimal value would be non-positive. Therefore, the upper bound is reduced substantially (with respect to sign change) from $\overline{x}_i > 0$ to a non-positive value.

The resulting linear program has less variables by $n - 1$ than the previous one from Sect. 2.1, and the number of constraints is the same. That is why the time complexities are almost the same.

3 Examples

The examples below show how optimal preconditoners behave for various initial enclosures, for various optimality criteria and for both versions (direct and residual) of the Gauss–Seidel method. For the residual form of the interval Gauss–Seidel iteration, we employed the minimal width approach (Sect. 2.1), and for the direct version, we used both the minimum upper and maximum lower bounds preconditioners.

The main purpose if the examples is to illustrate that while in some cases an optimal preconditioner makes no improvement, in another cases it may significantly reduce the overestimation. The computations were done in MATLAB with help of the interval toolbox INTLAB v7.1 (see Rump [16]).

Example 1. Consider Example 4 from Popova [12], where

$$A(p) := \begin{pmatrix} 1 & p_1 \\ p_1 & p_2 \end{pmatrix}, \quad b(p) := \begin{pmatrix} p_3 \\ p_3 \end{pmatrix}, \quad p \in \boldsymbol{p} = ([0, 1], -[1, 4], [0, 2])^T.$$

The initial enclosure of Σ_p is obtained by calling the `verifylss` function from Intlab on the relaxed system $A(\boldsymbol{p})x = b(\boldsymbol{p})$,

$$\boldsymbol{x} = ([-4.4849, 6.6667], [-5.3334, 4.9697])^T.$$

First, we call the residual form of the interval parametric Gauss–Seidel method. For the center $x^* := x^c$ and the residual interval vector $\boldsymbol{y} := \boldsymbol{x} - x^c$, one iteration yields the same result

$$\boldsymbol{y}^1 = ([-5.3940, 5.3940], [-4.1516, 4.1516])^T$$

for the respectively midpoint inverse and the minimal width preconditioners

$$(A^c)^{-1} = \begin{pmatrix} 0.9091 & 0.1818 \\ 0.1818 & -0.3636 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0.2 \\ 0.5 & -1 \end{pmatrix}.$$

The corresponding contracted enclosure is

$$\boldsymbol{x}^1 = ([-4.3031, 6.4850], [-4.3334, 3.9698])^T.$$

In contrast, the direct interval parametric Gauss–Seidel iteration with midpoint inverse preconditioning gives

$$\boldsymbol{x}^2 = ([-4.2668, 6.6668], [-4.3334, 3.3334])^T,$$

which tightened about 13.77 % of the interval width on average, whereas the optimal preconditioners yields

$$\boldsymbol{x}^3 = ([-4.0308, 6.6283], [-3.8769, 2.6812])^T,$$

which tightened about 20.38 % of the interval width on average. This enclosure was computed by calculating separately the upper and the lower end-points by using respectively the preconditioners

$$C^u = \begin{pmatrix} 1 & 0.2115 \\ 0.5978 & -1 \end{pmatrix}, \quad C^l = \begin{pmatrix} 1 & 0.1889 \\ 0.4022 & -1 \end{pmatrix}.$$

Comparing $\boldsymbol{x}^1$ and $\boldsymbol{x}^3$, we see that no one is better than the other one w.r.t. inclusion.

It is interesting to consider the interval hull of the relaxed system, $\boldsymbol{x}^4 = ([0, 4], [-2, 2])^T$, as an initial enclosure, too. For the residual form method, the midpoint inverse preconditioner does not improve this enclosure, but the optimal preconditioner reduces it to

$$\boldsymbol{x}^5 = ([0.0000, 3.8182], [-2.0001, 1.7686])^T.$$

For the direct version, the midpoint inverse preconditioner also fails to tighten $\boldsymbol{x}^4$, whereas the optimal preconditioner reduces the second component by half to

$$\boldsymbol{x}^6 = ([0, 4], [-2, 0])^T.$$

Example 2. In Example 5.2 from Popova and Krämer [15], a resistive network was considered with uncertain resistances. The output voltage was computed by solving the interval parametric system

$$A(p) := \begin{pmatrix} 30 & -10 & -10 & -10 & 0 \\ -10 & 10+p_1+p_2 & -p_1 & 0 & 0 \\ -10 & -p_1 & 15+p_1+p_3 & -5 & 0 \\ -10 & 0 & -5 & 15+p_4 & 0 \\ 0 & 0 & -5 & 5 & 1 \end{pmatrix}, \quad b(p) := \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where $p \in \boldsymbol{p} = [8,12] \times [4,8] \times [8,12] \times [8,12]$.

We will consider the enclosure computed by the residual and the direct interval Gauss–Seidel method with the inverse midpoint preconditioner and initiated by the `verifylss` enclosure for the relaxed system.

The residual form yields the enclosure

$$\boldsymbol{x}^1 = ([0.0595, 0.0851], [0.0262, 0.0587], [0.0247, 0.0514], [0.0251, 0.0479], [-0.0352, 0.0499])^T,$$

which is no further improved by the optimal preconditioner.

The direct form yields in $0.2194\,s$ the enclosure

$$\boldsymbol{x}^2 = ([0.0575, 0.0871], [0.0268, 0.0660], [0.0247, 0.0557], [0.0267, 0.0491], [-0.0527, 0.0674])^T.$$

Using the optimal preconditioner, it takes $1.2535\,s$ to reduce the enclosure radii by 15 % on average, and the resulting enclosure is

$$\boldsymbol{x}^3 = ([0.0626, 0.0862], [0.0293, 0.0646], [0.0273, 0.0541], [0.0276, 0.0482], [-0.0359, 0.0573])^T.$$

For comparison, `verifylss` enclosure for the system preconditioned by the inverse midpoint reads

$$\boldsymbol{x}^4 = ([0.0576, 0.0871], [0.0187, 0.0662], [0.0202, 0.0558], [0.0240, 0.0491], [-0.0525, 0.0672])^T.$$

Hence, our enclosure $\boldsymbol{x}^3$ has by about 22 % (on average) smaller radii than $\boldsymbol{x}^4$.

4 Conclusion

We proposed a linear programming based method to compute an optimal preconditioning matrix for the parametric interval Gauss–Seidel iterations. Even though large numerical studies would be needed, some illustrative examples show that the optimal preconditioner can sometimes reduce the ubiquitous overestimation. Besides that, future research may be addressed to other types of optimality

(S-preconditioners, pivoting preconditioners, and others), or to directly focus on the interval Newton method (as done in Kearfott [5,6,8]). It would be also interesting to investigate optimality of various preconditioners in generalized interval systems, for instance for AE solutions of (non)-parametric interval systems [14].

Acknowledgments. The author was supported by the Czech Science Foundation Grant P402-13-10660S.

References

1. Fiedler, M., Nedoma, J., Ramík, J., Rohn, J., Zimmermann, K.: Linear Optimization Problems with Inexact Data. Springer, New York (2006)
2. Gau, C.-Y., Stadtherr, M.A.: New interval methodologies for reliable chemical process modeling. Comput. Chem. Eng. **26**(6), 827–840 (2002)
3. Hladík, M.: Enclosures for the solution set of parametric interval linear systems. Int. J. Appl. Math. Comput. Sci. **22**(3), 561–574 (2012)
4. Hladík, M.: New operator and method for solving real preconditioned interval linear equations. SIAM J. Numer. Anal. **52**(1), 194–206 (2014)
5. Kearfott, R.B.: Preconditioners for the interval Gauss-Seidel method. SIAM J. Numer. Anal. **27**(3), 804–822 (1990)
6. Kearfott, R.B.: Decomposition of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems. Comput. **47**(2), 169–191 (1991)
7. Kearfott, R.B.: A comparison of some methods for bounding connected and disconnected solution sets of interval linear systems. Comput. **82**(1), 77–102 (2008)
8. Kearfott, R.B., Hu, C., Novoa, M.: A review of preconditioners for the interval Gauss-Seidel method. Interval Comput. **1991**(1), 59–85 (1991)
9. Lin, Y., Stadtherr, M.A.: Advances in interval methods for deterministic global optimization in chemical engineering. J. Glob. Optim. **29**(3), 281–296 (2004)
10. Neumaier, A.: New techniques for the analysis of linear interval equations. Linear Algebra Appl. **58**, 273–325 (1984)
11. Neumaier, A.: Interval Methods for Systems of Equations. Cambridge University Press, Cambridge (1990)
12. Popova, E.: Quality of the solution sets of parameter-dependent interval linear systems. ZAMM, Z. Angew. Math. Mech. **82**(10), 723–727 (2002)
13. Popova, E.D.: On the solution of parametrised linear systems. In: Krämer, W., von Gudenberg, J.W. (eds.) Scientific Computing, Validated Numerics, Interval Methods, pp. 127–138. Kluwer (2001)
14. Popova, E.D., Hladík, M.: Outer enclosures to the parametric AE solution set. Soft Comput. **17**(8), 1403–1414 (2013)
15. Popova, E.D., Krämer, W.: Visualizing parametric solution sets. BIT **48**(1), 95–115 (2008)
16. Rump, S.M.: INTLAB - INTerval LABoratory. In: Csendes, T. (ed.) Developments in Reliable Computing, pp. 77–104. Kluwer Academic Publishers, Dordrecht (1999)
17. Rump, S.M.: Verification methods: Rigorous results using floating-point arithmetic. Acta Numer. **19**, 287–449 (2010)