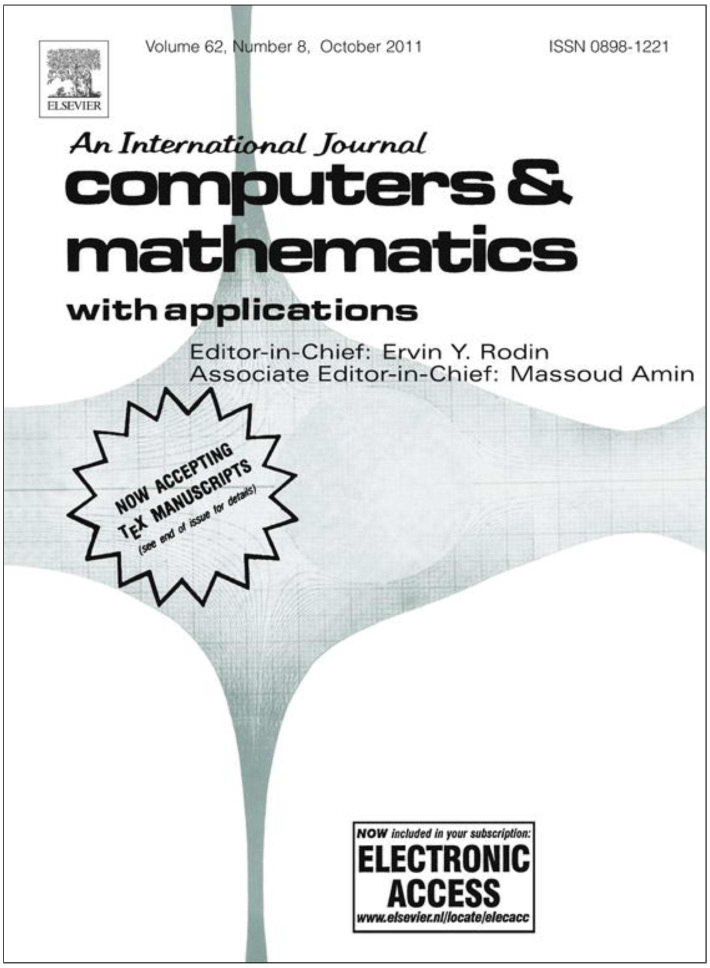
Volume 62, Number 8, October 2011
ISSN 0898-1221
ELSEVIER
An International Journal
computers & mathematics
with applications
Editor-in-Chief: Ervin Y. Rodin
Associate Editor-in-Chief: Massoud Amin
NOW ACCEPTING TEX MANUSCRIPTS
(see end of issue for details)
NOW included in your subscription:
ELECTRONIC ACCESS
www.elsevier.nl/locate/elecacc

Computers and Mathematics with Applications 62 (2011) 3152–3163

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Characterizing and approximating eigenvalue sets of symmetric interval matrices

Milan Hladík[a,b], David Daney[b], Elias Tsigaridas[c,*]

[a] Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic
[b] INRIA Sophia-Antipolis Méditerranée, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France
[c] Computer Science Department, Aarhus University, Denmark

ARTICLE INFO

Article history:
Received 23 March 2011
Received in revised form 26 June 2011
Accepted 8 August 2011

Keywords:
Interval matrix
Symmetric matrix
Interval analysis
Eigenvalue
Eigenvalue bounds

ABSTRACT

We consider the eigenvalue problem for the case where the input matrix is symmetric and its entries are perturbed, with perturbations belonging to some given intervals. We present a characterization of some of the exact boundary points, which allows us to introduce an inner approximation algorithm, that in many case estimates exact bounds. To our knowledge, this is the first algorithm that is able to guarantee exactness. We illustrate our approach by several examples and numerical experiments.

1. Introduction

Computing eigenvalues of a matrix is a basic linear algebraic task used throughout in mathematics, physics and computer science. Real life makes this problem more complicated by imposing uncertainties and measurement errors on the matrix entries. We suppose we are given some compact intervals in which the matrix entries can vary. The set of all possible real eigenvalues forms a compact set, and the question that we deal with in this paper is how to characterize and compute it.

The interval eigenvalue problem has its own history. The first results are probably due to Deif [1] and Rohn & Deif [2]: bounds for real and imaginary parts for complex eigenvalues were studied by Deif [1], while Rohn & Deif [2] considered real eigenvalues. Their theorems are applicable only under an assumption on sign pattern invariancy of eigenvectors, which is not easy to verify (cf. [3]). A boundary point characterization of the eigenvalue set was given by Rohn [4], and it was used by Hladík et al. [5] to develop a branch & prune algorithm producing an arbitrarily tight approximation of the eigenvalue set. Another approximate method was given by Qiu et al. [6]. The related topic of finding verified intervals of eigenvalues for real matrices was studied in, e.g. [7–9].

In this paper we consider the case of the symmetric eigenvalue problem. Symmetric matrices naturally appear in many practical problems, but symmetric interval matrices are hard to deal with. This is so, mainly due to the so-called dependencies, that is, correlations between the matrix components. If we "forget" these dependencies and solve the problem by reducing it to the previous case, then the results will be greatly overestimated, in general (but not the extremal points, see Theorem 2). From now on we consider only the symmetric case.

Due to the dependencies just mentioned, the theoretical background for the eigenvalue problem of symmetric interval matrices is not well established enough and there are few practical methods. The known results are by Deif [1] and Hertz [10].

[*] Corresponding author.
E-mail addresses: milan.hladik@matfyz.cz, milan.hladik@sophia.inria.fr (M. Hladík), david.daney@sophia.inria.fr (D. Daney), elias@cs.au.dk (E. Tsigaridas).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.08.028

Deif [1] gives an exact description of the eigenvalue set together with restrictive assumptions. Hertz [10] (cf. [11]) proposed a formula for computing two extremal points of the eigenvalue set—the largest and the smallest ones. As the problem itself is very hard, it is not surprising conjectures on the problem [12] turned out to be wrong [13].

In recent years, several approximation algorithms have been developed. By means of matrix perturbation theory, Qiu et al. [14] proposed an algorithm for approximate bounds, and Leng & He [15] for an outer estimation. An outer estimation was also considered by Kolev [16], but for the general case with nonlinear dependencies. Some initial bounds that are easy and quick to compute were discussed by Hladík et al. [17], and an iterative refinement in [18]. An iterative algorithm for outer estimation was given by Beaumont [19].

In this paper we focus more on the inner approximations (subsets) of the eigenvalue sets. There are much fewer papers devoted to inner approximation. Let us mention an evolution strategy method by Yuan et al. [13] or a general method for nonlinear systems [9].

The interval eigenvalue problem has a lot of applications in the field of mechanics and engineering. Let us mention for instance automobile suspension systems [6], mass structures [14], vibrating systems [20], principal component analysis [21], and robotics [22]. Another applications arise from the engineering area concerning singular values and condition numbers. Using the well-known Jordan–Wielandt transformation [23, Section 8.6], [24, Section 7.5] we can simply reduce a singular value calculation to a symmetric eigenvalue one.

This paper is organized as follows. In Section 2 we introduce the notation that we use throughout the paper. In Section 3 we present our main theoretical result that enables to exactly determine some of the eigenvalue set. It is a basis for the algorithms that we present in Section 4. The algorithms calculate inner approximations of the eigenvalue sets. Even though outer approximation is usually considered in literature, inner approximation is of interest, too. Moreover, due to the main theorem, we can obtain exact eigenvalue bounds in some cases. Finally, in Section 5 we demonstrate our approach by a number of examples and numerical experiments.

2. Basic definitions and theoretical background

Let us introduce some notions first. An interval matrix is denoted by boldface and defined as

$$\boldsymbol{A} := [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n};\ \underline{A} \le A \le \overline{A}\},$$

where $\underline{A},\ \overline{A} \in \mathbb{R}^{m\times n}$, $\underline{A} \le \overline{A}$, are given matrices. By

$$A_c := \frac{1}{2}(\underline{A} + \overline{A}), \qquad A_\Delta := \frac{1}{2}(\overline{A} - \underline{A})$$

we denote the midpoint and the radius of $\boldsymbol{A}$, respectively.

By an interval linear system of equations $\boldsymbol{A}x = \boldsymbol{b}$ we mean a family of systems $Ax = b$, such that $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$. In a similar way we introduce interval linear systems of inequalities and mixed systems of equations and inequalities. A vector x is a solution of $\boldsymbol{A}x = \boldsymbol{b}$ if it is a solution of $Ax = b$ for some $A \in \boldsymbol{A}$ and $b \in \boldsymbol{b}$. We assume that the reader is familiar with the basics of interval arithmetic; for further details we refer to e.g. [25–27].

Let $\mathcal{F}$ be a family of $n \times n$ matrices. We denote the eigenvalue set of the family $\mathcal{F}$ by

$$\Lambda(\mathcal{F}) := \{\lambda \in \mathbb{R};\ \exists A \in \mathcal{F}\ \exists x \ne 0 : Ax = \lambda x\}.$$

A symmetric interval matrix as defined as

$$\boldsymbol{A}^S := \{A \in \boldsymbol{A} \mid A = A^T\}.$$

It is usually a proper subset of $\boldsymbol{A}$. Considering the eigenvalue set $\Lambda(\boldsymbol{A})$, it generally represents an overestimation of $\Lambda(\boldsymbol{A}^S)$. That is why we focus directly on the eigenvalue set of the symmetric portion, even though we must take into account the dependencies between the elements, in the definition of $\boldsymbol{A}^S$.

A real symmetric matrix $A \in \mathbb{R}^{n\times n}$ has always n real eigenvalues, let us sort them in non-increasing order

$$\lambda_1(A) \ge \lambda_2(A) \ge \cdots \ge \lambda_n(A).$$

We extend this notation for symmetric interval matrices

$$\boldsymbol{\lambda}_i(\boldsymbol{A}^S) := \{\lambda_i(A) \mid A \in \boldsymbol{A}^S\}.$$

These sets represent n compact intervals $\boldsymbol{\lambda}_i(\boldsymbol{A}^S) = [\underline{\lambda}_i(\boldsymbol{A}^S), \overline{\lambda}_i(\boldsymbol{A}^S)]$, $i = 1, \ldots, n$; cf. [17]. The intervals can be disjoint, can overlap, or some of them, can be identical. However, what cannot happen is that one interval is a proper subset of another interval. The union of these intervals produces $\Lambda(\boldsymbol{A}^S)$. For instance, consider an interval matrix

$$\boldsymbol{A}^S = \begin{pmatrix} [2,3] & 0 \\ 0 & [1,4] \end{pmatrix}. \tag{1}$$

Then $\boldsymbol{\lambda}_1(\boldsymbol{A}^S) = [2, 4]$, $\boldsymbol{\lambda}_2(\boldsymbol{A}^S) = [1, 3]$ and $\Lambda(\boldsymbol{A}^S) = [1, 4]$.

Throughout the paper we use the following notation:
$\lambda_i(A)$ the ith eigenvalue of a symmetric matrix A (in non-increasing order)
$\sigma_i(A)$ the ith singular value of a matrix A (in non-increasing order)
$\upsilon_i(A)$ the eigenvector associated to the ith eigenvalue of a symmetric matrix A
$\rho(A)$ the spectral radius of a matrix A
$\partial\mathcal{S}$ the boundary of a set $\mathcal{S}$
conv $\mathcal{S}$ the convex hull of a set $\mathcal{S}$
diag(y) the diagonal matrix with entries $y_1, \dots, y_n$
sgn(x) the sign vector of a vector x, i.e., $\operatorname{sgn}(x) = (\operatorname{sgn}(x_1), \dots, \operatorname{sgn}(x_n))^T$
$\|x\|_2$ the Euclidean vector norm, i.e., $\|x\|_2 = \sqrt{x^Tx}$
$\|x\|_\infty$ the Chebyshev (maximum) vector norm, i.e., $\|x\|_\infty = \max\{|x|_i;\ i = 1, \dots, n\}$
$x \le y, A \le B$ vector and matrix relations are understood component-wise.

3. Main theorem

The following theorem is the main theoretical result of the present paper. We remind the reader that the principal $m \times m$ submatrix of a given $n \times n$ matrix is any submatrix obtained by eliminating any $n - m$ rows and the corresponding $n - m$ columns.

Theorem 1. *Let $\lambda \in \partial\Lambda(\boldsymbol{A}^S)$. Then there is $k \in \{1, \dots, n\}$ and a principal submatrix $\hat{\boldsymbol{A}}^S \subset \mathbb{R}^{k\times k}$ of $\boldsymbol{A}^S$ such that:*

- *If $\lambda = \overline{\lambda}_j(\boldsymbol{A}^S)$ for some $j \in \{1, \dots, n\}$, then*

$$\lambda \in \{\lambda_i(\hat{A}_c + \operatorname{diag}(z)\hat{A}_\Delta \operatorname{diag}(z));\ z \in \{\pm 1\}^k,\ i = 1, \dots, k\}. \tag{2}$$

- *If $\lambda = \underline{\lambda}_j(\boldsymbol{A}^S)$ for some $j \in \{1, \dots, n\}$, then*

$$\lambda \in \{\lambda_i(\hat{A}_c - \operatorname{diag}(z)\hat{A}_\Delta \operatorname{diag}(z));\ z \in \{\pm 1\}^k,\ i = 1, \dots, k\}. \tag{3}$$

Proof. Let $\lambda \in \partial\Lambda(\boldsymbol{A}^S)$. Then either $\lambda = \overline{\lambda}_j(\boldsymbol{A}^S)$ or $\lambda = \underline{\lambda}_j(\boldsymbol{A}^S)$, for some $j \in \{1, \dots, n\}$. We assume the former case. The latter can be proved similarly.

The eigenvalue λ corresponds to a matrix $A \in \boldsymbol{A}$. Without loss of generality we assume that the corresponding eigenvector x, $\|x\|_2 = 1$, is of the form $x = (0^T, y^T)^T$, where $y \in \mathbb{R}^k$ and $y_i \neq 0$, for all $1 \le i \le k$, and for some $k \in \{1, \dots, n\}$. The symmetric interval matrix $\boldsymbol{A}^S$ can be written as

$$\boldsymbol{A}^S = \begin{pmatrix} \boldsymbol{B}^S & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D}^S \end{pmatrix},$$

where $\boldsymbol{B}^S \subset \mathbb{R}^{(n-k)\times(n-k)}$, $\boldsymbol{C} \subset \mathbb{R}^{(n-k)\times k}$, $\boldsymbol{D}^S \subset \mathbb{R}^{k\times k}$. This can be achieved by a suitable permutation $P^T\boldsymbol{A}^SP$, where P is a permutation matrix. Notice that $P^T\boldsymbol{A}^SP$ remains symmetric with the same eigenvalues and eigenvectors, and no overestimation occurs since $P^T\boldsymbol{A}^SP$ has the same entries as $\boldsymbol{A}^S$ but at different positions.

From the basic equality $Ax = \lambda x$ it follows that

$$Cy = 0 \quad \text{for some } C \in \boldsymbol{C}, \tag{4}$$

and

$$Dy = \lambda y \quad \text{for some } D \in \boldsymbol{D}^S. \tag{5}$$

We focus on the latter relation; it says that λ is an eigenvalue of D. We will show that $\boldsymbol{D}^S$ is the required principal submatrix $\hat{\boldsymbol{A}}^S$ thanks to the proposed permutation, and D could be written as in (2).

From (5) we have that $\lambda = y^TDy$, and hence the partial derivatives are

$$\frac{\partial\lambda}{\partial d_{ij}} = y_iy_j \neq 0, \quad i, j = 1, \dots, k.$$

This relation strongly influences the structure of D. If $y_iy_j > 0$, then $d_{ij} = \overline{d}_{ij}$. This is so, because otherwise by increasing d_{ij} we also increase the value of λ, which contradicts our assumption that λ lies on the upper boundary of $\Lambda(\boldsymbol{A}^S)$. Likewise, $y_iy_j < 0$ implies $d_{ij} = \underline{d}_{ij}$. This allows us to write D in the following more compact form

$$D = D_c + \operatorname{diag}(z)D_\Delta \operatorname{diag}(z), \tag{6}$$

where $z = \operatorname{sgn}(y) \in \{\pm 1\}^k$. Therefore, λ belongs to a set as the one presented in the right-hand side of (2), which completes the proof. □

Note that not every $\underline{\lambda}_j(\boldsymbol{A}^S)$ or $\overline{\lambda}_j(\boldsymbol{A}^S)$ is a boundary point of $\Lambda(\boldsymbol{A}^S)$; see (1). Theorem 1 is also true for such $\underline{\lambda}_j(\boldsymbol{A}^S)$ or $\overline{\lambda}_j(\boldsymbol{A}^S)$ that are non-boundary, but represent no multiple eigenvalue (since the corresponding eigenvector is uniquely determined). However, correctness of Theorem 1 for all $\underline{\lambda}_j(\boldsymbol{A}^S)$ and $\overline{\lambda}_j(\boldsymbol{A}^S)$, $j = 1, \ldots, n$, is still an open question. Moreover, full characterization of all $\underline{\lambda}_j(\boldsymbol{A}^S)$ and $\overline{\lambda}_j(\boldsymbol{A}^S)$, $j = 1, \ldots, n$, is lacking too.

As we have already mentioned, in general, the eigenvalue set of an interval matrix is larger than the eigenvalue set of its symmetric portion. This is true even if both the midpoint and radius matrices are symmetric (see Example 1). The following theorem says that overestimation caused by the additional matrices is somehow limited by the convex hull area. An illustration will be given in Example 1, where

$$\Lambda(\boldsymbol{A}^S) = [3.7321,\ 6.7843] \cup [0.00000,\ 0.3230] \cup [-4.1072,\ -1.0000],$$
$$\Lambda(\boldsymbol{A}) = [3.7321,\ 6.7843] \cup [-0.6458,\ 0.3230] \cup [-4.1072,\ -1.0000].$$

The lower bounds and the upper bounds $\Lambda(\boldsymbol{A}^S)$ and $\Lambda(\boldsymbol{A})$ are always the same, but the other boundary points may differ.

Theorem 2. *Let $A_c, A_\Delta \in \mathbb{R}^{n\times n}$ be symmetric matrices. Then*

$$\operatorname{conv} \Lambda(\boldsymbol{A}^S) = \operatorname{conv} \Lambda(\boldsymbol{A}).$$

Proof. The inclusion $\operatorname{conv} \Lambda(\boldsymbol{A}^S) \subseteq \operatorname{conv} \Lambda(\boldsymbol{A})$ follows from the definition of the convex hull.

Let $A \in \boldsymbol{A}$ be arbitrary, λ one of its real eigenvalues, and x the corresponding eigenvector, where $\|x\|_2 = 1$. Let $B := \frac{1}{2}(A + A^T) \in \boldsymbol{A}^S$, then the following holds:

$$\lambda = x^T Ax \leq \max_{\|y\|_2=1} y^T Ay = \max_{\|y\|_2=1} y^T By = \lambda_1(B) \in \operatorname{conv} \Lambda(\boldsymbol{A}^S).$$

Similarly,

$$\lambda = x^T Ax \geq \min_{\|y\|_2=1} y^T Ay = \min_{\|y\|_2=1} y^T By = \lambda_n(B) \in \operatorname{conv} \Lambda(\boldsymbol{A}^S).$$

Therefore $\lambda \in \operatorname{conv} \Lambda(\boldsymbol{A}^S)$, and so $\operatorname{conv} \Lambda(\boldsymbol{A}) \subseteq \operatorname{conv} \Lambda(\boldsymbol{A}^S)$, which completes the proof. □

4. Inner approximation algorithms

Theorem 1 naturally yields an algorithm to compute a very sharp inner approximation of $\Lambda(\boldsymbol{A}^S)$, which could also be exact in some cases. We will present the algorithm in the sequel (Section 4.3). First, we define some notions and propose two simple but useful methods for inner approximations.

Any subset of $\mathcal{S}$ is called an *inner approximation*. Similarly, any set that contains $\mathcal{S}$ is called an *outer approximation*. In our case, an inner approximation of the eigenvalue set $\boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, is denoted by $\boldsymbol{\mu}_i(\boldsymbol{A}^S) = [\underline{\mu}_i(\boldsymbol{A}^S), \overline{\mu}_i(\boldsymbol{A}^S)] \subseteq \boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, and an outer approximation is denoted by $\boldsymbol{\omega}_i(\boldsymbol{A}^S) = [\underline{\omega}_i(\boldsymbol{A}^S), \overline{\omega}_i(\boldsymbol{A}^S)] \supseteq \boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, where $1 \leq i \leq n$.

From a practical point of view, an outer approximation is usually more useful. However, an inner approximation is also important in some applications. For example, it could be used to measure quality (sharpness) of an outer approximation, or it could be used to prove the (Hurwitz or Schur) instability of certain interval matrices, cf. [28].

We introduce three inner approximation algorithms. The first one, a local improvement, is an efficient algorithm, but needn't be very accurate. On the contrary, vertex enumeration gives more accurate results (two bounds are exact), but it is more costly. Eventually, submatrix vertex enumeration yields the tightest inner approximation but on the account of the time complexity.

4.1. Local improvement

The first algorithm that we present is based on a local improvement search technique. A similar method, but for interval matrices $\boldsymbol{A}$ with A_c and A_Δ symmetric, was proposed by Rohn [28]. The basic idea of the algorithm is to start with an eigenvalue, $\lambda_i(A_c)$, and the corresponding eigenvector, $v_i(A_c)$, of the midpoint matrix, A_c, and then move to an extremal matrix in $\boldsymbol{A}^S$ according to the sign pattern of the eigenvector. The procedure is repeated until no improvement is possible.

Algorithm 1 outputs the upper boundaries $\overline{\mu}_i(\boldsymbol{A}^S)$ of the inner approximation $[\underline{\mu}_i(\boldsymbol{A}^S), \overline{\mu}_i(\boldsymbol{A}^S)]$, where $1 \leq i \leq n$. The lower boundaries, $\underline{\mu}_i(\boldsymbol{A}^S)$, can be obtained similarly. The validity of the procedure follows from the fact that every considered matrix, A, belongs to $\boldsymbol{A}^S$.

The algorithm terminates after at most $2^{n-1} + 1$ iterations since we can normalize $v_i(A)$ such that the first entry is non-negative. However, usually in practice the number of iterations is much smaller, which makes the algorithm attractive for applications. Our numerical experiments (Section 5) indicate that the number of iterations is rarely greater than two, even for matrices of dimension 20. Moreover, the resulting inner approximation is quite sharp, depending on the width of intervals in $\boldsymbol{A}^S$. This is not surprising as whenever the input intervals are narrow enough, the algorithm produces, sometimes

Algorithm 1 (Local improvement for $\overline{\mu}_i(\mathbf{A}^S)$, $i = 1, \ldots, n$)

```
1:  for $i = 1, \ldots, n$ do
2:    $\overline{\mu}_i(\mathbf{A}^S) = -\infty$;
3:    $A := A_c$;
4:    while $\lambda_i(A) > \overline{\mu}_i(\mathbf{A}^S)$ do
5:      $\overline{\mu}_i(\mathbf{A}^S) := \lambda_i(A)$;
6:      $D := \mathrm{diag}(\mathrm{sgn}(v_i(A)))$;
7:      $A := A_c + DA_\Delta D$;
8:    end while
9:  end for
10: return $\overline{\mu}_i(\mathbf{A}^S),\ i = 1, \ldots, n$.
```

even after the first iteration, exact bounds; see [1]. This is due to sign invariancy of eigenvectors, which enables to set up an optimal scenario in steps 6 and 7. If the eigenvectors have no invariant signs of their entries, then we still can achieve the optimal bound by the local improvement.

We refer the reader to Section 5 for a more detailed presentation of the experiments.

4.2. Vertex enumeration

The second method that we present is based on enumeration of some special boundary matrices of $\mathbf{A}$. It consists of inspecting all matrices

$$A_z := A_c + \mathrm{diag}(z)A_\Delta \mathrm{diag}(z), \quad z \in \{\pm 1\}^n, \quad z_1 = 1, \tag{7}$$

and continuously improving an inner approximation $\overline{\mu}_i(\mathbf{A}^S)$, whenever $\lambda_i(A_z) > \overline{\mu}_i(\mathbf{A}^S)$, where $1 \le i \le n$. The lower bounds, $\underline{\mu}_i(\mathbf{A}^S)$, could be obtained in a similar way using the matrices $A_c - \mathrm{diag}(z)A_\Delta \mathrm{diag}(z)$, where $z \in \{\pm 1\}^n$, and $z_1 = 1$. The condition $z_1 = 1$ follows from the fact that $\mathrm{diag}(z)A_\Delta \mathrm{diag}(z) = \mathrm{diag}(-z)A_\Delta \mathrm{diag}(-z)$, which gives us the freedom to fix one component of z. The number of steps that the algorithm performs is 2^{n-1}. Therefore, this method is suitable only for matrices of moderate dimensions.

The main advantages of the *vertex enumeration* approach are the following. First, it provides us with a sharper inner approximation of the eigenvalue sets than the local improvement; in local improvement we inspect only some of the matrices in (7). Second, two of the computed bounds are exact; by Hertz [10] (cf. [11]) and Hertz [29] we have that $\overline{\mu}_1(\mathbf{A}^S) = \overline{\lambda}_1(\mathbf{A}^S)$ and $\underline{\mu}_n(\mathbf{A}^S) = \underline{\lambda}_n(\mathbf{A}^S)$. Concerning the other bounds calculated by vertex enumeration, even though it was conjectured that there were exact [12], it turned out that they were not exact, in general [13]. The assertion by Hertz [29, Theorem 1] that $\underline{\mu}_1(\mathbf{A}^S) = \underline{\lambda}_1(\mathbf{A}^S)$ and $\overline{\mu}_n(\mathbf{A}^S) = \overline{\lambda}_n(\mathbf{A}^S)$ is wrong, too; see Example 3. Nevertheless, Theorem 1 and its proof indicate a sufficient condition: if no eigenvector corresponding to an eigenvalue of $\mathbf{A}^S$ has a zero component, then the vertex enumeration yields exactly the eigenvalue sets $\boldsymbol{\lambda}_i(\mathbf{A}^S)$, $i = 1, \ldots, n$. This is easy to see from the proof of Theorem 1; the submatrices in question is only the matrix $\mathbf{A}$ itself, and the values (2)–(3) correspond to matrices that are processed by vertex enumeration.

The efficient implementation of this approach is quite challenging. In order to overcome in practice the exponential complexity of the algorithm, we implemented a branch & bound algorithm, which is in accordance with the suggestions of Rohn [28]. However, the adopted bounds are not that tight, and the actual running times are usually worse than the direct vertex enumeration; it is probably because of weak pruning part of the exhaustive search, so one has to go through almost all the search tree. That is why we do not consider further this variant. The direct vertex enumeration scheme for computing the upper bounds, $\overline{\mu}_i(\mathbf{A}^S)$, is presented in Algorithm 2.

Algorithm 2 (Vertex enumeration for $\overline{\mu}_i(\mathbf{A}^S)$, $i = 1, \ldots, n$)

```
for $i = 1, \ldots, n$ do
  $\overline{\mu}_i(\mathbf{A}^S) = \lambda_i(A_c)$;
end for
for all $z \in \{\pm 1\}^n,\ z_1 = 1,$ do
  $A := A_c + \mathrm{diag}(z)A_\Delta \mathrm{diag}(z)$;
  for $i = 1, \ldots, n$ do
    if $\lambda_i(A) > \overline{\mu}_i(\mathbf{A}^S)$ then
      $\overline{\mu}_i(\mathbf{A}^S) := \lambda_i(A)$;
    end if
  end for
end for
return $\overline{\mu}_i(\mathbf{A}^S),\ i = 1, \ldots, n$.
```

4.3. Submatrix vertex enumeration

In this section we present an algorithm that is based on Theorem 1, and it usually produces very tight inner approximations, even exact ones in some cases. The basic idea underlying the algorithm is to enumerate all the vertices of all the principal submatrices of $\boldsymbol{A}^S$ including $\boldsymbol{A}^S$ itself. Thus we go through more matrices than vertex enumeration and the method yields more accurate approximation, but with higher time complexity. The number of steps performed with this approach is

$$2^{n-1} + n2^{n-2} + \binom{n}{2} 2^{n-2} + \cdots + n2^0 = \frac{1}{2}(3^n - 1).$$

To overcome the obstacle of the exponential number of iterations, at least in practice, we notice that not all eigenvalues of the principal submatrices of the matrices in $\boldsymbol{A}^S$ belong to some of the eigenvalue sets $\boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, where $1 \le i \le n$. For this we will introduce a condition for checking such an inclusion.

Assume that we are given an inner approximation $\boldsymbol{\mu}_i(\boldsymbol{A}^S)$ and an outer approximation $\boldsymbol{\omega}_i(\boldsymbol{A}^S)$ of the eigenvalue sets $\boldsymbol{\lambda}_i(\boldsymbol{A}^S)$; that is $\boldsymbol{\mu}_i(\boldsymbol{A}^S) \subseteq \boldsymbol{\lambda}_i(\boldsymbol{A}^S) \subseteq \boldsymbol{\omega}_i(\boldsymbol{A}^S)$, where $1 \le i \le n$. As we will see in the sequel, the quality of the output of our methods depends naturally on the sharpness of the outer approximation used.

Let $\boldsymbol{D}^S \subset \mathbb{R}^{k\times k}$ be a principal submatrix of $\boldsymbol{A}^S$ and, without loss of generality, assume that it is situated in the right-bottom corner, i.e.,

$$\boldsymbol{A}^S = \begin{pmatrix} \boldsymbol{B}^S & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D}^S \end{pmatrix},$$

where $\boldsymbol{B}^S \subset \mathbb{R}^{(n-k)\times(n-k)}$ and $\boldsymbol{C} \subset \mathbb{R}^{(n-k)\times k}$. This can be obtained by an appropriate permutation $P^T\boldsymbol{A}^S P$, where P is a permutation matrix as in the proof of Theorem 1.

Let λ be an eigenvalue of some vertex matrix $D \in \boldsymbol{D}^S$, which is of the form (6), and let y be the corresponding eigenvector. If the eigenvector is not unique then λ is a multiple eigenvalue and therefore it is a simple eigenvalue of some principal submatrix of $\boldsymbol{D}^S$ due to Cauchy's interlacing property for eigenvalues [23, Theorem 8.1.7] [24, Example 7.5.3]; in this case we restrict our consideration to this submatrix.

Let $p \in \{1, \dots, n\}$ be fixed. We want to determine whether λ is equal to $\overline{\lambda}_p(\boldsymbol{A}^S) \in \Lambda(\boldsymbol{A}^S)$, or, if this is not possible, to improve the upper bound $\overline{\mu}_p(\boldsymbol{A}^S)$; the lower bound can be handled accordingly. In view of (4), $Cy = 0$ must hold for some $C \in \boldsymbol{C}$, whence

$$0 \in \boldsymbol{C}y.$$

So λ is an eigenvalue of some matrix in $\boldsymbol{A}^S$. Now, we are sure that $\lambda \in \Lambda(\boldsymbol{A}^S)$ and it remains to determine whether λ also belongs to $\boldsymbol{\lambda}_p(\boldsymbol{A}^S)$.

If $\lambda \le \overline{\mu}_p(\boldsymbol{A}^S)$, then it is useless to further consider λ, since it would not improve the inner approximation of the pth eigenvalue set. Suppose $\lambda > \overline{\mu}_p(\boldsymbol{A}^S)$. If $p = 1$ or $\lambda < \underline{\omega}_{p-1}(\boldsymbol{A}^S)$, then λ must belong to $\boldsymbol{\lambda}_p(\boldsymbol{A}^S)$, and we can improve the inner bound $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda$. In this case the algorithm terminates early, and that is the reason we need $\boldsymbol{\omega}_i(\boldsymbol{A}^S)$, $1 \le i \le n$, to be as tight as possible.

If $p > 1$ and $\lambda \ge \underline{\omega}_{p-1}(\boldsymbol{A}^S)$, we proceed as follows. We pick an arbitrary $C \in \boldsymbol{C}$, such that $Cy = 0$; we refer to, e.g. [30] for details on the selection process. Next, we select an arbitrary $B \in \boldsymbol{B}^S$ and let

$$A := \begin{pmatrix} B & C \\ C^T & D \end{pmatrix}. \tag{8}$$

We compute the eigenvalues of A, and if $\overline{\mu}_p(\boldsymbol{A}^S) < \lambda_p(A)$, then we set $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda_p(A)$, otherwise we do nothing.

However, it can happen that $\lambda = \overline{\lambda}_i(\boldsymbol{A}^S)$, and we do not identify it, and hence we do not enlarge the inner estimation $\mu_p(\boldsymbol{A}^S)$. Nevertheless, if we apply the method for all $p = 1, \dots, n$ and all principal submatrices of $\boldsymbol{A}^S$, then we touch all the boundary points of $\Lambda(\boldsymbol{A}^S)$. If $\lambda \in \partial\Lambda(\boldsymbol{A}^S)$, then λ is covered by the resulting inner approximation. In the case when λ is an upper boundary point, we consider the maximal $i \in \{1, \dots, n\}$ such that $\lambda = \overline{\lambda}_i(\boldsymbol{A}^S)$ and then the ith eigenvalue of the matrix (8) must be equal to λ. Similar tests are valid for a lower boundary point.

Now we have all the ingredients at hand for the direct version of the submatrix vertex enumeration approach that is presented in Algorithm 3, which improves the upper bound $\overline{\mu}_p(\boldsymbol{A}^S)$ of an inner approximation, where the index p is still fixed. Let us also mention that in step 4 of Algorithm 3, the decomposition of $\boldsymbol{A}^S$ according to the index set J means that $\boldsymbol{D}^S$ is a restriction of $\boldsymbol{A}^S$ to the rows and the columns indexed by J, $\boldsymbol{B}^S$ is a restriction of $\boldsymbol{A}^S$ to the rows and the columns indexed by $\{1, \dots, n\} \setminus J$, and $\boldsymbol{C}$ is a restriction of $\boldsymbol{A}^S$ to the rows indexed by $\{1, \dots, n\} \setminus J$ and the columns indexed by J.

4.3.1. Branch & bound improvement

In order to tackle the exponential worst case complexity of Algorithm 3, we propose the following modification. Instead of inspecting all non-empty subsets of $\{1, \dots, n\}$ in step 3, we exploit a branch & bound method, which may skip some useless subsets. Let a non-empty $J \subseteq \{1, \dots, n\}$ be given. The new, possibly improved, eigenvalue λ must lie in the interval

Algorithm 3 (Direct submatrix vertex enumeration for $\overline{\mu}_p(\boldsymbol{A}^S)$)

```
compute outer approximation $\boldsymbol{\omega}_i(\boldsymbol{A}^S)$, $i = 1, \dots, n$;
call Algorithm 1 to get inner approximation $\boldsymbol{\mu}_i(\boldsymbol{A}^S)$, $i = 1, \dots, n$;
for all $J \subseteq \{1, \dots, n\}$, $J \neq \emptyset$, do
  decompose $\boldsymbol{A}^S = \begin{pmatrix} \boldsymbol{B}^S & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D}^S \end{pmatrix}$ according to $J$;
  for all $z \in \{\pm 1\}^{|J|}$, $z_1 = 1$, do
    $D := D_c + \mathrm{diag}(z) D_\Delta \mathrm{diag}(z)$;
    for $i = 1, \dots, |J|$ do
      $\lambda := \lambda_i(D)$;
      $y := v_i(D)$;
      if $\lambda > \overline{\mu}_p(\boldsymbol{A}^S)$ and $\lambda \le \overline{\omega}_p(\boldsymbol{A}^S)$ and $0 \in \boldsymbol{C}y$ then
        if $p = 1$ or $\lambda < \underline{\omega}_{p-1}(\boldsymbol{A}^S)$ then
          $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda$;
        else
          find $C \in \boldsymbol{C}$ such that $Cy = 0$;
          $A := \begin{pmatrix} B_c & C \\ C^T & D \end{pmatrix}$;
          if $\lambda_p(A) > \overline{\mu}_p(\boldsymbol{A}^S)$ then
            $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda_p(A)$;
          end if
        end if
      end if
    end for
  end for
end for
return $\overline{\mu}_p(\boldsymbol{A}^S)$.
```

$\boldsymbol{\lambda} := [\,\overline{\mu}_p(\boldsymbol{A}^S), \overline{\omega}_p(\boldsymbol{A}^S)]$. If this is the case, then the interval matrix $\boldsymbol{A}^S - \boldsymbol{\lambda} I$ must be irregular, i.e., it contains a singular matrix. Moreover, the interval system

$$(\boldsymbol{A}^S - \boldsymbol{\lambda} I)x = 0, \quad \|x\|_\infty = 1,$$

has a solution x, where $x_i = 0$ for all $i \notin J$. We decompose $\boldsymbol{A}^S - \boldsymbol{\lambda} I$ according to J, and, without loss of generality, we may assume that $J = \{n - |J| + 1, \dots, n\}$, then

$$\boldsymbol{A}^S - \boldsymbol{\lambda} I = \begin{pmatrix} \boldsymbol{B}^S - \boldsymbol{\lambda} I & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D}^S - \boldsymbol{\lambda} I \end{pmatrix}.$$

The interval system becomes

$$\boldsymbol{C}y = 0, \qquad (\boldsymbol{D}^S - \boldsymbol{\lambda} I)y = 0, \quad \|y\|_\infty = 1, \tag{9}$$

where we considered $x = (0^T, y^T)^T$. This is a very useful necessary condition. If (9) has no solution, then we cannot improve the current inner approximation. We can also prune the whole branch with J as a root; that is, we will inspect no index sets $J' \subseteq J$. The strength of this condition follows from the fact that the system (9) is overconstrained, it has more equations than variables. Therefore, with high probability that it has no solution, even for larger J.

Let us make two comments about the interval system (9). First, this system has a lot of dependencies. They are caused from the multiple occurrences of $\boldsymbol{\lambda}$, and by the symmetry of $\boldsymbol{D}^S$. If no solver for interval systems that can handle dependencies is available, then we can solve (9) as an ordinary interval system, "forgetting" the dependencies. The necessary condition will be weaker, but still valid. This is what we did in our implementation.

The second comment addresses the expression $\|y\|_\infty = 1$. We have chosen the maximum norm in order that the interval system be linear. The expression could be rewritten as $-1 \le y \le 1$ (for checking solvability of (9) we can use either normalization $\|y\|_\infty = 1$ or $\|y\|_\infty \le 1$). Another possibility is to write

$$-1 \le y \le 1, \quad y_i = 1 \quad \text{for some } i \in \{1, \dots, |J|\}.$$

This indicates that we can split the problem into solving $|J|$ interval systems

$$\boldsymbol{C}y = 0, \qquad (\boldsymbol{D}^S - \boldsymbol{\lambda} I)y = 0, \quad -1 \le y \le 1, \quad y_i = 1,$$

where i runs, sequentially, through all the values $\{1, \dots, |J|\}$; cf. the ILS method proposed in [5]. The advantage of this approach is that the overconstrained interval systems have (one) more equation than the original overconstrained system, and hence the resulting necessary condition could be stronger. Our numerical results discussed in Section 5 concern this variant. As a solver for interval systems we utilize the convex approximation approach by Beaumont [31]; it is sufficiently fast and produces narrow enough approximations of the solution set.

4.3.2. How to conclude for exact bounds?

Let us summarize properties of the submatrix vertex enumeration method. On the one hand the worst case complexity of the algorithm is rather prohibitive, $O(3^n)$, but on the other hand, we obtain better inner approximations, and sometimes we get exact bounds of the eigenvalue sets. Theorem 1 and the discussion in the previous section allow us to recognize exact bounds. Namely, for any $i \in \{2, \dots, n\}$, we have that if $\overline{\lambda}_i(\boldsymbol{A}^S) < \underline{\lambda}_{i-1}(\boldsymbol{A}^S)$, then $\overline{\mu}_i(\boldsymbol{A}^S) = \overline{\lambda}_i(\boldsymbol{A}^S)$; a similar inequality holds for the lower bound. This is a rather theoretical recipe because we may not know a priori whether the assumption is satisfied. However, we can propose a sufficient condition: if $\overline{\omega}_i(\boldsymbol{A}^S) < \underline{\omega}_{i-1}(\boldsymbol{A}^S)$, then two successive eigenvalue sets do not overlap and the assumption is obviously true. In this case we conclude $\overline{\mu}_i(\boldsymbol{A}^S) = \overline{\lambda}_i(\boldsymbol{A}^S)$; otherwise we cannot conclude.

This sufficient condition is another reason why we need a sharp outer approximation. The sharper it is, the more often we are able to conclude that the exact bound is achieved.

Exploiting the condition we can also decrease the running time of submatrix vertex enumeration. We call Algorithm 3 only for $p \in \{1, \dots, n\}$ such that $p = 1$ or $\overline{\omega}_p(\boldsymbol{A}^S) < \underline{\omega}_{p-1}(\boldsymbol{A}^S)$. The resulting inner approximation may be a bit less tight, but the number of exact boundary points of $\varLambda(\boldsymbol{A}^S)$ that we can identify remains the same.

Notice that there is enough open space for developing better conditions. For instance, we do not know whether $\overline{\mu}_i(\boldsymbol{A}^S) < \underline{\mu}_{i-1}(\boldsymbol{A}^S)$ (computed by submatrix vertex enumeration) can serve also as a sufficient condition for the purpose of determining exact bounds.

5. Numerical experiments

In this section we present some examples and numerical results illustrating properties of the proposed algorithms. We performed the experiments on a PC `Intel(R) Core 2`, CPU 3 GHz, 2 GB RAM, and the source code was written in C++. We use `GLPK v.4.23` [32] for solving linear programs, `CLAPACK v.3.1.1` for its linear algebraic routines, and `PROFIL/BIAS v.2.0.4` [33] for interval arithmetic and basic operations. Notice, however, that routines of `GLPK` and `CLAPACK` [34] do not produce verified solutions; for real-life problems this may not be acceptable.

Example 1. Consider the following symmetric interval matrix

$$\boldsymbol{A}^S = \begin{pmatrix} 1 & 2 & [1,5] \\ 2 & 1 & 1 \\ [1,5] & 1 & 1 \end{pmatrix}^S .$$

Local improvement (Algorithm 1) yields an inner approximation

$$\boldsymbol{\mu}_1(\boldsymbol{A}^S) = [3.7321,\ 6.7843],$$
$$\boldsymbol{\mu}_2(\boldsymbol{A}^S) = [0.0888,\ 0.3230],$$
$$\boldsymbol{\mu}_3(\boldsymbol{A}^S) = [-4.1072,\ -1.0000].$$

The same result is obtained by the vertex enumeration (Algorithm 2). Therefore, $\overline{\mu}_1(\boldsymbol{A}^S) = \overline{\lambda}_1(\boldsymbol{A}^S)$ and $\underline{\mu}_3(\boldsymbol{A}^S) = \underline{\lambda}_3(\boldsymbol{A}^S)$. An outer approximation that is needed by the submatrix vertex enumeration (Algorithm 3) is computed using the methods of Hladík et al. [17,18]. It is

$$\boldsymbol{\omega}_1(\boldsymbol{A}^S) = [3.5230,\ 6.7843],$$
$$\boldsymbol{\omega}_2(\boldsymbol{A}^S) = [0.0000,\ 1.0519],$$
$$\boldsymbol{\omega}_3(\boldsymbol{A}^S) = [-4.1214,\ -0.2019].$$

Now, the submatrix vertex enumeration algorithm yields the inner approximation

$$\boldsymbol{\mu}'_1(\boldsymbol{A}^S) = [3.7321,\ 6.7843],$$
$$\boldsymbol{\mu}'_2(\boldsymbol{A}^S) = [0.0000,\ 0.3230],$$
$$\boldsymbol{\mu}'_3(\boldsymbol{A}^S) = [-4.1072,\ -1.0000].$$

Since the outer approximation intervals do not overlap, we can conclude that this approximation is exact, that is, $\boldsymbol{\lambda}_i(\boldsymbol{A}^S) = \boldsymbol{\mu}'_i(\boldsymbol{A}^S)$, $i = 1, 2, 3$.

This example shows two important aspects of the interval eigenvalue problem. First, it demonstrates that the vertex enumeration does not produce exact bounds in general. Second, the symmetric eigenvalue set can be a proper subset of the unsymmetric one, i.e., $\Lambda(\boldsymbol{A}^S) \subsetneq \Lambda(\boldsymbol{A})$. This could be easily seen by the matrix

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 5 & 1 & 1 \end{pmatrix}.$$

It has three real eigenvalues 4.6458, -0.6458 and -1.0000, but the second one does not belong to $\Lambda(\boldsymbol{A}^S)$. Indeed, using the method by Hladík et al. [5] we obtain

$$\Lambda(\boldsymbol{A}) = [3.7321,\ 6.7843] \cup [-0.6458,\ 0.3230] \cup [-4.1072,\ -1.0000].$$

Example 2. Consider the example given by Qiu et al. [14] (see also [17,13]):

$$\boldsymbol{A}^S = \begin{pmatrix} [2975, 3025] & [-2015, -1985] & 0 & 0 \\ [-2015, -1985] & [4965, 5035] & [-3020, -2980] & 0 \\ 0 & [-3020, -2980] & [6955, 7045] & [-4025, -3975] \\ 0 & 0 & [-4025, -3975] & [8945, 9055] \end{pmatrix}^S.$$

The local improvement (Algorithm 1) yields an inner approximation

$$\boldsymbol{\mu}_1(\boldsymbol{A}^S) = [12560.8377,\ 12720.2273], \qquad \boldsymbol{\mu}_2(\boldsymbol{A}^S) = [7002.2828,\ 7126.8283],$$
$$\boldsymbol{\mu}_3(\boldsymbol{A}^S) = [3337.0785,\ 3443.3127], \qquad \boldsymbol{\mu}_4(\boldsymbol{A}^S) = [842.9251,\ 967.1082].$$

The vertex enumeration (Algorithm 2) produces the same result. Hence we can state that $\overline{\mu}_1(\boldsymbol{A}^S)$ and $\underline{\mu}_4(\boldsymbol{A}^S)$ are optimal.

To call the last method, submatrix vertex enumeration (Algorithm 3) we need an outer approximation. We use the following by [17]

$$\boldsymbol{\omega}_1(\boldsymbol{A}^S) = [12560.6296,\ 12720.2273], \qquad \boldsymbol{\omega}_2(\boldsymbol{A}^S) = [6990.7616,\ 7138.1800],$$
$$\boldsymbol{\omega}_3(\boldsymbol{A}^S) = [3320.2863,\ 3459.4322], \qquad \boldsymbol{\omega}_4(\boldsymbol{A}^S) = [837.0637,\ 973.1993].$$

Now, submatrix vertex enumeration yields the same inner approximation as the previous methods. However, now we have more information. Since the outer approximation intervals are mutually disjoint, the obtained results are the best possible. Therefore, $\boldsymbol{\mu}_i(\boldsymbol{A}^S) = \boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, where $i = 1, \ldots, 4$.

Example 3. Herein, we present two examples for approximating the singular values of an interval matrix. Let $A \in \mathbb{R}^{m\times n}$ and $q := \min\{m, n\}$. By the Jordan–Wielandt theorem [23, Section 8.6], [24, Section 7.5] the singular values $\sigma_1(A) \geq \cdots \geq \sigma_q(A)$ of A are identical to the q largest eigenvalues of the symmetric matrix

$$\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}.$$

Thus, if we consider the singular value sets $\boldsymbol{\sigma}_1(\boldsymbol{A}), \ldots, \boldsymbol{\sigma}_q(\boldsymbol{A})$ of some interval matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, we can identify them as the q largest eigenvalue sets of the symmetric interval matrix

$$\boldsymbol{M} := \begin{pmatrix} 0 & \boldsymbol{A}^T \\ \boldsymbol{A} & 0 \end{pmatrix}^S.$$

(1) Consider the following interval matrix from [35]

$$\boldsymbol{A} = \begin{pmatrix} [2, 3] & [1, 1] \\ [0, 2] & [0, 1] \\ [0, 1] & [2, 3] \end{pmatrix}.$$

Both the local improvement and the vertex enumeration result in the same inner approximation, i.e.

$$\boldsymbol{\mu}_1(\boldsymbol{M}) = [2.5616,\ 4.5431], \qquad \boldsymbol{\mu}_2(\boldsymbol{M}) = [1.2120, 2.8541].$$

Thus, $\overline{\sigma}_1(\boldsymbol{A}) = 4.5431$. Additionally, consider the following outer approximation from [17].

$$\boldsymbol{\omega}_1(\boldsymbol{M}) = [2.0489,\ 4.5431], \qquad \boldsymbol{\omega}_2(\boldsymbol{M}) = [0.4239,\ 3.1817].$$

Using Algorithm 3, we obtain

$$\boldsymbol{\mu}_1'(\boldsymbol{M}) = [2.5616,\ 4.5431], \qquad \boldsymbol{\mu}_2'(\boldsymbol{M}) = [1.0000, 2.8541].$$

Now we can claim that $\underline{\sigma}_2(\boldsymbol{A}) = 1$, since $\underline{\omega}_2(\boldsymbol{M}) > 0$. Unfortunately, we cannot conclude about the exact values of the remaining quantities, since the two outer approximation intervals overlap. We only know that $\underline{\sigma}_1(\boldsymbol{A}) \in [2.0489,\ 2.5616]$ and $\overline{\sigma}_2(\boldsymbol{A}) \in [2.8541,\ 3.1817]$.

(2) The second example comes from Ahn & Chen [36]. Let $\boldsymbol{A}$ be the following interval matrix

$$\boldsymbol{A} = \begin{pmatrix} [0.75,\ 2.25] & [-0.015,\ -0.005] & [1.7,\ 5.1] \\ [3.55,\ 10.65] & [-5.1,\ -1.7] & [-1.95,\ -0.65] \\ [1.05,\ 3.15] & [0.005,\ 0.015] & [-10.5,\ -3.5] \end{pmatrix}.$$

Both local improvement and vertex enumeration yield the same result, i.e.

$$\boldsymbol{\mu}_1(\boldsymbol{M}) = [4.6611,\ 13.9371], \qquad \boldsymbol{\mu}_2(\boldsymbol{M}) = [2.2140,\ 11.5077],$$
$$\boldsymbol{\mu}_3(\boldsymbol{M}) = [0.1296,\ 2.9117].$$

Hence, $\overline{\sigma}_1(\boldsymbol{A}) = 13.9371$. As an outer approximation we use the following intervals calculated by a method from [17]

$$\boldsymbol{\omega}_1(\boldsymbol{M}) = [4.3308,\ 14.0115], \qquad \boldsymbol{\omega}_2(\boldsymbol{M}) = [1.9305,\ 11.6111],$$
$$\boldsymbol{\omega}_3(\boldsymbol{M}) = [0.0000,\ 5.1000].$$

Running the submatrix vertex enumeration, we get the inner approximation

$$\boldsymbol{\mu}'_1(\boldsymbol{M}) = [4.5548,\ 13.9371], \qquad \boldsymbol{\mu}'_2(\boldsymbol{M}) = [2.2140,\ 11.5077],$$
$$\boldsymbol{\mu}'_3(\boldsymbol{M}) = [0.1296,\ 2.9517].$$

We cannot conclude that $\underline{\sigma}_3(\boldsymbol{A}) = \underline{\mu}_3(\boldsymbol{A}) = 0.1296$, because $\boldsymbol{\omega}_3(\boldsymbol{M})$ has a nonempty intersection with the fourth largest eigenvalue set, which is equal to zero. Also the other singular value sets remain uncertain, but within the computed inner and outer approximations.

Notice that $\underline{\mu}'_1(\boldsymbol{M}) < \underline{\mu}_1(\boldsymbol{M})$, whence $\underline{\mu}'_1(\boldsymbol{M}) < \underline{\lambda}_1(\boldsymbol{M}) = \underline{\sigma}_1(\boldsymbol{A})$ disproving the Hertz's theorem 1 from [29] that the lower and upper limits of $\boldsymbol{\lambda}_1(\boldsymbol{M})$ and $\boldsymbol{\lambda}_n(\boldsymbol{M})$ are computable by the vertex enumeration method. It is true only for $\overline{\lambda}_1(\boldsymbol{M})$ and $\underline{\lambda}_n(\boldsymbol{M})$.

Example 4. In this example we present some randomly generated examples of large dimensions. The entries of the midpoint matrix, A_c, are taken randomly in $[-20, 20]$ using the uniform distribution. The entries of the radius matrix A_Δ are taken randomly, using the uniform distribution in $[0, R]$, where R is a positive real number. We applied our algorithm on the interval matrix $\boldsymbol{M} := \boldsymbol{A}^T\boldsymbol{A}$, because it has a convenient distribution of eigenvalue set—some are overlapping and some are not. Sharpness of results is measured using the quantity

$$1 - \frac{e^T \mu_\Delta(\boldsymbol{M}^S)}{e^T \omega_\Delta(\boldsymbol{M}^S)},$$

where $e = (1, \dots, 1)^T$. This quantity lies always within the interval $[0, 1]$. The closer to zero it is, the tighter the approximation. In addition, if it is zero, then we achieved exact bounds for every eigenvalue set $\boldsymbol{\lambda}_i(\boldsymbol{M}^S)$, $1 \le i \le n$. The initial outer approximation, $\boldsymbol{\omega}_i(\boldsymbol{M}^S)$, $1 \le i \le n$, was computed using the method due of Hladík et al. [17], and filtered by the method proposed by Hladík et al. in [18]. Finally, it was refined according to the comment in Section 4.3.2. For the submatrix vertex enumeration algorithm we implemented the branch & bound improvement, which is described in Sections 4.3.1 and 4.3.2.

The results are displayed in Table 1; the values are appropriately rounded. We see that local improvement yields almost as tight inner approximation as vertex enumeration, but with much lower effort. Submatrix vertex enumeration is even more costly, but it can sometimes conclude for exact bounds, so the approximation is more accurate, particularly for narrow input intervals.

Example 5. In this example we present some numerical results on approximating singular value sets as introduced in Example 3. The input consists of an interval (rectangular) matrix $\boldsymbol{A} \subset \mathbb{R}^{m\times n}$ which is selected randomly as in the previous example.

Table 2 presents our experiments. The time in the table corresponds to the computation of the approximation of only the q largest eigenvalue sets of the Jordan–Wielandt matrix. The behavior of the three algorithms is similar to that in Example 4.

6. Conclusion and future directions

We proposed a new solution theorem for the symmetric interval eigenvalue problem, which describes some of the boundary points of the eigenvalue set. Unfortunately, the complete characterization is still a challenging open problem.

We developed an inner approximation algorithm (submatrix vertex enumeration), which in the case where the eigenvalue sets are disjoint, and the intermediate gaps are wide enough, outputs exact results. To our knowledge, even under this assumption, this is the first algorithm that can guarantee exact bounds. Thus, it can be used in correspondence with outer approximation methods to produce exact eigenvalue sets.

We carried out comparisons with other inner approximation methods, local improvement and vertex enumeration. The local improvement method is very efficient with sufficiently tight bounds. The vertex enumeration is more time

Table 1
Eigenvalues of random interval symmetric matrices $\boldsymbol{A}^T\boldsymbol{A}$ of dimension $n \times n$.

n	R	Algorithm 1 (local improvement)		Algorithm 2 (vertex enumeration)		Algorithm 3 (submatrix vertex enumeration)	
		Sharpness	Time (s)	Sharpness	Time	Sharpness	Time
5	0.001	0.05817	0.00	0.05041	0.00 s	0.00000	0.04 s
5	0.01	0.07020	0.00	0.05163	0.00 s	0.00000	0.03 s
5	0.1	0.26273	0.00	0.23389	0.00 s	0.17332	0.04 s
5	1	0.25112	0.00	0.23644	0.00 s	0.20884	0.01 s
10	0.001	0.08077	0.00	0.07412	0.09 s	0.00000	1.15 s
10	0.01	0.13011	0.01	0.11982	0.08 s	0.04269	1.29 s
10	0.1	0.27378	0.01	0.25213	0.09 s	0.12756	3.17 s
10	1	0.56360	0.01	0.52330	0.09 s	0.52256	2.58 s
15	0.001	0.07991	0.02	0.07557	7.3 s	0.00000	16.47 s
15	0.01	0.21317	0.02	0.19625	6.5 s	0.11341	2 min 29 s
15	0.1	0.36410	0.02	0.34898	7.0 s	0.34869	4 min 58 s
15	1	0.76036	0.02	0.73182	7.2 s	0.73182	7.5 s
20	0.001	0.09399	0.06	0.09080	7 min 21 s	0.00000	13 min 46 s
20	0.01	0.24293	0.06	0.22976	7 min 6 s	0.12574	1 h 14 min 55 s
20	0.1	0.43199	0.06	0.40857	7 min 14 s	0.22360	1 h 15 min 41 s
20	1	0.82044	0.06	0.79967	7 min 33 s	0.79967	7 min 39 s
25	0.001	0.14173	0.13	0.13397	6 h 53 min 0 s	0.02871	9 h 32 min 54 s

Table 2
Singular values of random interval matrices of dimension $m \times n$.

m	n	R	Algorithm 1 (local improvement)		Algorithm 2 (vertex enumeration)		Algorithm 3 (submatrix vertex enumeration)	
			Sharpness	Time (s)	Sharpness	Time	Sharpness	Time
5	5	0.01	0.08945	0.00	0.07716	0.10 s	0.00000	0.53 s
5	5	0.1	0.09876	0.01	0.09270	0.08 s	0.00000	0.73 s
5	5	1	0.43560	0.01	0.31419	0.10 s	0.26795	4.34 s
5	10	0.01	0.11320	0.02	0.10337	5.79 s	0.00000	7.91 s
5	10	0.1	0.13032	0.02	0.12321	5.98 s	0.00000	8.40 s
5	10	1	0.35359	0.02	0.33176	5.52 s	0.22848	21.53 s
5	15	0.01	0.10603	0.05	0.09424	5 min 31 s	0.00000	5 min 36 s
5	15	0.1	0.17303	0.04	0.16758	5 min 33 s	0.00000	7 min 58 s
5	15	1	0.46064	0.05	0.39708	5 min 32 s	0.31847	15 min 47 s
10	10	0.01	0.10211	0.06	0.09652	8 min 3 s	0.00000	8 min 19 s
10	10	0.1	0.13712	0.07	0.13387	8 min 10 s	0.00000	14 min 12 s
10	10	1	0.39807	0.07	0.35580	7 min 52 s	0.30279	26 h 48 min 38 s
10	15	0.01	0.09561	0.12	0.09116	5 h 51 min 53 s	0.00000	5 h 54 min 56 s

consuming with slightly more accurate bounds, two of which are exact. Our numerical experiments suggest that the local search algorithm is superior to the other methods as long as the input matrices have higher dimension. However, for small dimensional problems with possibly narrow input intervals, the submatrix vertex enumeration approach gives very accurate bounds in reasonable time. Thus local improvement is suitable for high dimensional problems or for problems where computing time is important. Contrary, submatrix vertex enumeration is a good choice when accuracy is the main objective.

Acknowledgments

The authors thank the reviewers for their detailed and helpful comments.

ET is partially supported by an individual postdoctoral grant from the Danish Agency for Science, Technology and Innovation, and also acknowledges support from the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed.

References

[1] A.S. Deif, The interval eigenvalue problem, Z. Angew. Math. Mech. 71 (1) (1991) 61–64.
[2] J. Rohn, A. Deif, On the range of eigenvalues of an interval matrix, Computing 47 (3–4) (1992) 373–377.
[3] A. Deif, J. Rohn, On the invariance of the sign pattern of matrix eigenvectors under perturbation, Linear Algebr. Appl. 196 (1994) 63–70.
[4] J. Rohn, Interval matrices: singularity and real eigenvalues, SIAM J. Matrix Anal. Appl. 14 (1) (1993) 82–91.
[5] M. Hladík, D. Daney, E.P. Tsigaridas, An algorithm for addressing the real interval eigenvalue problem, J. Comput. Appl. Math. 235 (8) (2011) 2715–2730.
[6] Z. Qiu, P.C. Müller, A. Frommer, An approximate method for the standard interval eigenvalue problem of real non-symmetric interval matrices, Comm. Numer. Methods Engrg. 17 (4) (2001) 239–251.
[7] G. Alefeld, G. Mayer, Interval analysis: theory and applications, J. Comput. Appl. Math. 121 (1–2) (2000) 421–464.

[8] S. Miyajima, T. Ogita, S. Rump, S. Oishi, Fast verification for all eigenpairs in symmetric positive definite generalized eigenvalue problems, Reliab. Comput. 14 (2010) 24–45.

[9] S.M. Rump, Verification methods: rigorous results using floating-point arithmetic, Acta Numer. 19 (2010) 287–449.

[10] D. Hertz, The extreme eigenvalues and stability of real symmetric interval matrices, IEEE Trans. Automat. Control 37 (4) (1992) 532–535.

[11] J. Rohn, A handbook of results on interval linear problems, 2005. http://www.cs.cas.cz/rohn/handbook.

[12] Z. Qiu, X. Wang, Solution theorems for the standard eigenvalue problem of structures with uncertain-but-bounded parameters, J. Sound Vib. 282 (1–2) (2005) 381–399.

[13] Q. Yuan, Z. He, H. Leng, An evolution strategy method for computing eigenvalue bounds of interval matrices, Appl. Math. Comput. 196 (1) (2008) 257–265.

[14] Z. Qiu, S. Chen, I. Elishakoff, Bounds of eigenvalues for structures with an interval description of uncertain-but-non-random parameters, Chaos Solitons Fractals 7 (3) (1996) 425–434.

[15] H. Leng, Z. He, Computing eigenvalue bounds of structures with uncertain-but-non-random parameters by a method based on perturbation theory, Comm. Numer. Methods Engrg. 23 (11) (2007) 973–982.

[16] L.V. Kolev, Outer interval solution of the eigenvalue problem under general form parametric dependencies, Reliab. Comput. 12 (2) (2006) 121–140.

[17] M. Hladík, D. Daney, E.P. Tsigaridas, Bounds on real eigenvalues and singular values of interval matrices, SIAM J. Matrix Anal. Appl. 31 (4) (2010) 2116–2129.

[18] M. Hladík, D. Daney, E.P. Tsigaridas, A filtering method for the interval eigenvalue problem, Appl. Math. Comput. 217 (12) (2011) 5236–5242.

[19] O. Beaumont, An algorithm for symmetric interval eigenvalue problem, Tech. Rep. IRISA-PI-00-1314, Institut de recherche en informatique et systèmes aléatoires, Rennes, France, 2000.

[20] A.D. Dimarogonas, Interval analysis of vibrating systems, J. Sound Vib. 183 (4) (1995) 739–749.

[21] F. Gioia, C.N. Lauro, Principal component analysis on interval data, Comput. Statist. 21 (2) (2006) 343–363.

[22] D. Chablat, P. Wenger, F. Majou, J. Merlet, An interval analysis based study for the design and the comparison of 3-dof parallel kinematic machines, Int. J. Robot. Res. 23 (6) (2004) 615–624.

[23] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, 1996.

[24] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, Philadelphia, 2000.

[25] G. Alefeld, J. Herzberger, Introduction to Interval Computations, Academic Press, London, 1983.

[26] E. Hansen, G.W. Walster, Global Optimization Using Interval Analysis, 2nd ed., Marcel Dekker, New York, 2004 (revised and expanded).

[27] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, Cambridge, 1990.

[28] J. Rohn, An algorithm for checking stability of symmetric interval matrices, IEEE Trans. Automat. Control 41 (1) (1996) 133–136.

[29] D. Hertz, Interval analysis: eigenvalue bounds of interval matrices, in: C.A. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization, Springer, New York, 2009, pp. 1689–1696.

[30] J. Rohn, Solvability of systems of interval linear equations and inequalities, in: M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, K. Zimmermann (Eds.), Linear Optimization Problems with Inexact Data, Springer, New York, 2006, pp. 35–77 (Chapter 2).

[31] O. Beaumont, Solving interval linear systems with linear programming techniques, Linear Algebr. Appl. 281 (1–3) (1998) 293–309.

[32] A. Makhorin, GLPK—GNU Linear Programming Kit. http://www.gnu.org/software/glpk/.

[33] O. Knüppel, D. Husung, C. Keil, PROFIL/BIAS — a C++ class library. http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html.

[34] CLAPACK — Linear Algebra PACKage written for C. http://www.netlib.org/clapack/.

[35] A.S. Deif, Singular values of an interval matrix, Linear Algebr. Appl. 151 (1991) 125–133.

[36] H.-S. Ahn, Y.Q. Chen, Exact maximum singular value calculation of an interval matrix, IEEE Trans. Automat. Control 52 (3) (2007) 510–514.