Characterizing and approximating eigenvalue sets of symmetric interval matrices

Milan Hladik, David Daney, Elias Tsigaridas

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Characterizing and approximating eigenvalue sets of symmetric interval matrices

Milan Hladík — David Daney — Elias Tsigaridas

N° 7544

Février 2011

Robotics

ISSN 0249-6399 ISRN INRIA/RR--7544--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SOPHIA ANTIPOLIS - MÉDITERRANÉE

Characterizing and approximating eigenvalue sets of symmetric interval matrices

Milan Hladík* † , David Daney† , Elias Tsigaridas‡

Theme : Robotics
Perception, Cognition, Interaction
Équipe-Projet Coprin

Rapport de recherche n° 7544 — Février 2011 — 19 pages

Abstract: We consider the eigenvalue problem for the case where the input matrix is symmetric and its entries perturb in 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.

Key-words: Interval matrix, symmetric matrix, interval analysis, eigenvalue, eigenvalue bounds

* Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic, e-mail: milan.hladik@matfyz.cz

† INRIA Sophia-Antipolis Méditerranée, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France, e-mail: FirstName.LastName(AT)inria.fr

‡ Computer Science Department, Aarhus University, Denmark. e-mail: elias@cs.au.dk

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Caractérisation et approximation des ensembles de valeurs propres des matrices symétriques intervalles

Résumé : We consider the eigenvalue problem for the case where the input matrix is symmetric and its entries perturb in 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.

Mots-clés : Matrice d'intervalles, matrice symétrique, analyse par intervalles, valeur propre, borne sur les valeurs propres

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 perturb. 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 [10] and Rohn & Deif [34]: bounds for real and imaginary parts for complex eigenvalues were studied by Deif [10], while Rohn & Deif [34] considered real eigenvalues. Their theorems are applicable only under an assumption on sign pattern invariancy of eigenvectors, which is not easy to verify (cf. [8]). A boundary point characterization of eigenvalue set was given by Rohn [30], and it was used by Hladík et al. [17] to develop a branch & prune algorithm producing an arbitrarily tight approximation of the eigenvalue set. Another approximate method was given by Qiu et al. [28]. The related topic of finding verified intervals of eigenvalues for real matrices was studied in e.g. [4].

In this paper we focus on 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 wide enough and there are few practical methods. The known results are that by Deif [10] and Hertz [15]. Deif [10] gives an exact description of the eigenvalue set together with restrictive its assumptions. Hertz [15] (cf. [32]) proposed a formula for computing two extremal points of the eigenvalue set—the largest and the smallest one. As the problem itself is very hard, it is not surprising conjectures on the problem [29] turned out to be wrong [35].

In the recent years, several approximation algorithms have been developed. An evolution strategy method by Yuan et al. [35] yields an inner approximation of the eigenvalue set. By means of matrix perturbation theory, Qiu et al. [27] proposed an algorithm for approximate bounds, and Leng & He [23] for an outer estimation. An outer estimation was also considered by Kolev [22], but for general case with nonlinear dependencies. Some initial bounds that are easy and fast to compute were discussed by Hladík et al. [18]. An iterative algorithm for outer estimation was given by Beaumont [6].

This problem has a lot of applications in the field of mechanics and engineering. Let us mention for instance automobile suspension systems [28], mass structures [27], vibrating systems [11], principal component analysis [12], and robotics [7]. Another applications arise from the engineering area concerning singular values and condition numbers. Using the well-known Jordan–Wielandt transformation [13, 20, 25] we can simply reduce a singular value calculation to a symmetric eigenvalue one.

The rest of the paper is structured as follows: In Sec. 2 we introduce the notation that we use throughout the paper. In Sec. 3 we present our main theoretical result, and in Sec. 4 we develop our algorithms for the problem. Finally, in Sec. 5 we demonstrate our approach by a number of examples and numerical experiments, and we conclude in Sec. 6.

2 Basic definitions and theoretical background

Let us introduce some notions first. An interval matrix is defined as

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

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

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

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. [3, 14, 26].

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

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

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 eigenvalue set of the symmetric counterpart, 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 a non-increasing order

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

We extend this notation for symmetric interval matrices

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

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. [18]. The intervals can be disjoint, can overlap, or some of them, can be identical. However, what it can not happen is one interval to be a proper subset of another interval. The union of these intervals produces $\Lambda(\boldsymbol{A}^S)$.

Throughout the paper we use the following notation:

$\lambda_i(A)$	the ith eigenvalue of a symmetric matrix A (in a non-increasing order)
$\sigma_i(A)$	the ith singular value of a matrix A (in a non-increasing order)
$v_i(A)$	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 $\mathcal{S}$
$\operatorname{conv}\mathcal{S}$	the convex hull of a set $\mathcal{S}$
$\operatorname{diag}(y)$	the diagonal matrix with entries $y_1, \ldots, y_n$
$\operatorname{sgn}(x)$	the sign vector of a vector x, i.e., $\operatorname{sgn}(x) = (\operatorname{sgn}(x_1), \ldots, \operatorname{sgn}(x_n))^T$
$\|x\|_2$	the Euclidean vector norm, i.e., $\|x\|_2 = \sqrt{x^T x}$
$\|x\|_\infty$	the Chebyshev (maximum) vector norm, i.e., $\|x\|_\infty = \max\{\|x\|_i;\ i = 1, \ldots, n\}$
$x \le y$, $A \le B$	vector and matrix relations are understood component-wise

3 Main theorem

The following theorem is the main theoretical result of the 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 same $n - m$ columns.

Theorem 1. *Let $\lambda \in \partial\Lambda(\boldsymbol{A}^S)$. There is a principal submatrix $\widetilde{\boldsymbol{A}}^S \in \mathbb{R}^{k\times k}$ of $\boldsymbol{A}^S$ such that:*

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

$$\lambda \in \left\{\lambda_i\big(\widetilde{A}_c + \operatorname{diag}(z)\,\widetilde{A}_\Delta \operatorname{diag}(z)\big);\ z \in \{\pm 1\}^k,\ i = 1, \ldots, k\right\}. \qquad (1)$$

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

$$\lambda \in \left\{\lambda_i\big(\widetilde{A}_c - \operatorname{diag}(z)\,\widetilde{A}_\Delta \operatorname{diag}(z)\big);\ z \in \{\pm 1\}^k,\ i = 1, \ldots, k\right\}. \qquad (2)$$

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 $j \in \{1, \ldots, n\}$. We assume the former case. The latter could be proved similarly.

The eigenvalue λ is achieved for a matrix $A \in \boldsymbol{A}$. It is without loss of generality to 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 $k \in \{1, \ldots, n\}$. 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 $\boldsymbol{B}^S \subset \mathbb{R}^{(n-k)\times(n-k)}$, $\boldsymbol{C} \subset \mathbb{R}^{(n-k)\times k}$, $\boldsymbol{D}^S \subset \mathbb{R}^{k\times k}$.

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

$$Cy = 0 \ \text{ for some } C \in \boldsymbol{C}, \qquad (3)$$

and

$$Dy = \lambda y \quad \text{for some } D \in \boldsymbol{D}^S. \tag{4}$$

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 $\widetilde{\boldsymbol{A}}^S$ and D could be written as in (1).

From (4) we have that $\lambda = y^T Dy$, and hence the partial derivatives are

$$\frac{\partial \lambda}{\partial d_{ij}} = y_i y_j \neq 0, \quad i, j = 1, \ldots, k.$$

This relation strongly influences the structure of D. If $y_i y_j > 0$, 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, $y_i y_j < 0$ implies $d_{ij} = \underline{d}_{ij}$. This allows us to write D in the following more compact form

$$D = D_c + \operatorname{diag}(z)\, D_\Delta \operatorname{diag}(z), \tag{5}$$

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 (1), 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 $\Lambda(\boldsymbol{A}^S)$. 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 make no multiple eigenvalue (since the corresponding eigenvector is uniquely determined). However, truthfulness 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 counterpart. 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 intermediate area.

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 (Hurwitz or Schur) unstability of certain interval matrices, cf. [31].

4.1 Local improvement

The first algorithm that we present is based on local improvement search technique. A similar method, but for another class of symmetric interval matrices was proposed by Rohn [31]. 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 $\boldsymbol{A}^S$.

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

```
for i = 1, ..., n do
  μ̄_i(A^S) = −∞;
  A := A_c;
  while λ_i(A) > μ̄_i(A^S) do
    D := diag(sgn(v_i(A)));
    A := A_c + D A_Δ D;
    μ̄_i(A^S) := λ_i(A);
  end while
end for
return μ̄_i(A^S), i = 1, ..., n.
```

The algorithm terminates after, at most, 2^n iterations. 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 even after the first iteration, exact bounds; see [10]. 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 vertex enumeration. It consists of inspecting all matrices

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

and continuously improving an inner approximation $\overline{\mu}_i(\boldsymbol{A}^S)$, whenever $\lambda_i(A_z) > \overline{\mu}_i(\boldsymbol{A}^S)$, where $1 \le i \le n$. The lower bounds, $\underline{\mu}_i(\boldsymbol{A}^S)$, could be obtained in a similar way using the matrices $A_c - \operatorname{diag}(z)\, A_\Delta \operatorname{diag}(z)$, where $z \in \{\pm 1\}^n$, and $z_1 = 1$. The condition $z_1 = 1$ follows from the fact that $\operatorname{diag}(z)\, A_\Delta \operatorname{diag}(z) = \operatorname{diag}(-z)\, A_\Delta \operatorname{diag}(-z)$, 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 sharper inner approximation of the eigenvalue sets than the local improvement. Second, two of the computed bounds are exact; by Hertz [15] (cf. [32]) and Hertz [16] we have that $\overline{\mu}_1(\boldsymbol{A}^S) = \overline{\lambda}_1(\boldsymbol{A}^S)$ and $\underline{\mu}_n(\boldsymbol{A}^S) = \underline{\lambda}_n(\boldsymbol{A}^S)$. The other bounds calculated by the vertex enumeration, even though it was conjectured that there were exact [29], it turned out that they were not exact, in general [35] The assertion by Hertz [16, Theorem 1] that $\underline{\mu}_1(\boldsymbol{A}^S) = \underline{\lambda}_1(\boldsymbol{A}^S)$ and $\overline{\mu}_n(\boldsymbol{A}^S) = \overline{\lambda}_n(\boldsymbol{A}^S)$ is wrong, too; see Example 3. Nevertheless, Theorem 1 and its proof indicate a sufficient condition: If no eigenvector corresponding to an eigenvalue of $\boldsymbol{A}^S$ has a zero component, then the vertex enumeration yields exactly the eigenvalue sets $\boldsymbol{\lambda}_i(\boldsymbol{A}^S)$, $i = 1, \ldots, n$.

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 [31]. However, the adopted bounds are not that tight, and the actual running times are usually worse than the direct vertex enumeration. That is why we do not consider further this variant. The direct vertex enumeration scheme for computing the upper bounds, $\overline{\mu}_i(\boldsymbol{A}^S)$, is presented in Algorithm 2.

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 the algorithm is to enumerate all the vertices of all the principal submatrices of $\boldsymbol{A}^S$. The number of steps performed with this approach is

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

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.

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

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

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 $1 \leq i \leq n$. 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 $\boldsymbol{C} \subset \mathbb{R}^{(n-k)\times k}$. Let λ be an eigenvalue of some vertex matrix $D \in \boldsymbol{D}^S$ which is of the form (5), 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 $\boldsymbol{D}^S$; in this case we restrict our consideration to this submatrix.

We want to determine whether λ is equal to $\overline{\lambda}_p(\boldsymbol{A}^S) \in \Lambda(\boldsymbol{A}^S)$ for some fixed $p \in \{1, \ldots, n\}$, 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 (3), it must be

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

so that λ to be 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 $\lambda \leq \overline{\mu}_p(\boldsymbol{A}^S)$, then it is useless to further considering λ, as it will not extend the inner approximation of the pth eigenvalue set. If $p = 1$ or $\lambda < \underline{\omega}_{p-1}(\boldsymbol{A}^S)$, then λ must belong to $\overline{\lambda}_p(\boldsymbol{A}^S)$, 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 $\boldsymbol{\omega}_i(\boldsymbol{A}^S)$ to be as tight as possible, where $1 \leq i \leq n$.

If $p > 1$ and $\lambda \geq \underline{\omega}_{p-1}(\boldsymbol{A}^S)$, we proceed as follows. We pick an arbitrary $C \in \boldsymbol{C}$, such that $Cy = 0$; we refer to, e.g. [33] for details on the selection process. Next, we select an arbitrary $B \in \boldsymbol{B}^S$ and let

$$A := \begin{pmatrix} B & C \\ C^T & D \end{pmatrix}. \tag{6}$$

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 $p = 1, \ldots, n$ 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 $i \in \{1, \ldots, n\}$ such that $\lambda = \overline{\lambda}_i(\boldsymbol{A}^S)$ and then the ith eigenvalue of the matrix (6) must be equal to λ. Similarly test 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 $\{1, \ldots, n\} \setminus J$, and $\boldsymbol{C}$ is a restriction of $\boldsymbol{A}^S$ to the rows indexed by $\{1, \ldots, n\} \setminus J$ and the columns indexed by J.

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

1: compute outer approximation $\boldsymbol{\omega}_i(\boldsymbol{A}^S)$, $i = 1, \ldots, n$;
2: call Algorithm 1 to get inner approximation $\boldsymbol{\mu}_i(\boldsymbol{A}^S)$, $i = 1, \ldots, n$;
3: **for all** $J \subseteq \{1, \ldots, n\},\ J \neq \emptyset$, **do**
4: decompose $\boldsymbol{A}^S = \begin{pmatrix} \boldsymbol{B}^S & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D}^S \end{pmatrix}$ according to J;
5: **for all** $z \in \{\pm 1\}^{|J|},\ z_1 = 1$, **do**
6: $D := D_c + \operatorname{diag}(z)\, D_\Delta \operatorname{diag}(z)$;
7: **for** $i = 1, \ldots, |J|$ **do**
8: $\lambda := \lambda_i(D)$;
9: $y := v_i(D)$;
10: **if** $\lambda > \overline{\mu}_p(\boldsymbol{A}^S)$ and $\lambda \leq \overline{\omega}_p(\boldsymbol{A}^S)$ and $0 \in \boldsymbol{C}y$ **then**
11: **if** $p = 1$ or $\lambda < \underline{\omega}_{p-1}(\boldsymbol{A}^S)$ **then**
12: $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda$;
13: **else**
14: find $C \in \boldsymbol{C}$ such that $Cy = 0$;
15: $A := \begin{pmatrix} B_c & C \\ C^T & D \end{pmatrix}$;
16: **if** $\lambda_p(A) > \overline{\mu}_p(\boldsymbol{A}^S)$ **then**
17: $\overline{\mu}_p(\boldsymbol{A}^S) := \lambda_p(A)$;
18: **end if**
19: **end if**
20: **end if**
21: **end for**
22: **end for**
23: **end for**
24: **return** $\overline{\mu}_p(\boldsymbol{A}^S)$.

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, possible improved, eigenvalue λ must lie in the interval $\boldsymbol{\lambda} := [\overline{\mu}_p(\boldsymbol{A}^S), \overline{\omega}_p(\boldsymbol{A}^S)]$. 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

$$\boldsymbol{C}y = 0, \ (\boldsymbol{D}^S - \boldsymbol{\lambda} I)y = 0, \ \|y\|_\infty = 1, \tag{7}$$

where we considered $x = (0^T, y^T)^T$. This is a very useful necessary condition. If (7) 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 (7) 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 (7). 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 (7) 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 to pertain linearity of the interval system. The expression could be rewritten as $-1 \le y \le 1$ (for checking solvability of (7) we can use either normalization $\|y\|_\infty = 1$ or $\|y\|_\infty \le 1$). Another possibility is to write

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

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

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

where i runs, sequentially, through all the values $\{1,\dots,|J|\}$; cf. the ILS method proposed in [17]. 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 [5]; 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, 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. 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 the assumption is obviously true, and 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 times we are able to conclude that the exact bound is achieved.

Exploiting the condition we can also decrease 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 $\Lambda(\boldsymbol{A}^S)$ 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. The experiments we performed 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` [24] for solving linear programs, `CLAPACK v.3.1.1` for its linear algebraic routines, and `PROFIL/BIAS v.2.0.4` [21] for interval arithmetic and basic operations. Notice, however, that routines of `GLPK` and `CLAPACK`[1] 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

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

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. [18, 19]. It is

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

Now, the submatrix vertex enumeration algorithm yields the inner approximation

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

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., $\Lambda(\boldsymbol{A}^S) \subsetneqq \Lambda(\boldsymbol{A})$. This could be easily seen in 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. [17] 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. [27] (see also [18, 35]):

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

The local improvement (Algorithm 1) yields an inner approximation

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

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 [18]

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

Now, submatrix vertex enumeration yields the same inner approximation as the previous methods. However, now we have more information. Since the outer approximation interval 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 [13, 20, 25] the singular values, $\sigma_1(A) \geq \cdots \geq \sigma_q(A)$, of A are identical with 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 [9].

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

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

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

Thus, $\overline{\sigma}_1(\boldsymbol{A}) = 4.5431$. Additionally, consider the following outer approximation from [18].

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

Using Algorithm 3, we obtain

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

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 know only 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 [2]. 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.

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

Hence, $\overline{\sigma}_1(\boldsymbol{A}) = 13.9371$. As an outer approximation we use the following intervals, using [18].

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

Running the submatrix vertex enumeration, we get the inner approximation

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

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 approximation.

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 [16] 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 $e = (1, \ldots, 1)^T$. 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. The initial outer approximation, $\boldsymbol{\omega}_i(\boldsymbol{M}^S)$, where $1 \leq i \leq n$, was computed using the method due of Hladík et al. [18], and filtered by the method proposed by Hladík et al. in [19]. 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 Section 4.3.1 and 4.3.2.

The results are displayed in Table 1.

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 $\boldsymbol{A} \subseteq \mathbb{R}^{m\times n}$ 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.

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

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

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

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

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 characterisation 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, output exact results. To our knowledge, even under this assumption, this is the first algorithm that can guarantee exact bounds.

Based on our numerical experiments suggest that the local search algorithm is superior to the submatrix vertex enumeration algorithm when the input matrices are not of very small dimension.

Acknowledgment. ET is partially supported by an individual postdoctoral grant from the Danish Agency for Science, Technology and Innovation.

References

[1] CLAPACK – Linear Algebra PACKage written for C. `http://www.netlib.org/clapack/`.

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

[3] G. Alefeld and J. Herzberger. *Introduction to interval computations.* Academic Press, London, 1983.

[4] G. Alefeld and G. Mayer. Interval analysis: Theory and applications. *J. Comput. Appl. Math.*, 121(1-2):421–464, 2000.

[5] O. Beaumont. Solving interval linear systems with linear programming techniques. *Linear Algebra and Its Applications*, 281(1-3):293–309, 1998.

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

[7] D. Chablat, P. Wenger, F. Majou, and 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):615–624, 2004.

[8] A. Deif and J. Rohn. On the invariance of the sign pattern of matrix eigenvectors under perturbation. *Linear Algebra Appl.*, 196:63–70, 1994.

[9] A. S. Deif. Singular values of an interval matrix. *Linear Algebra Appl.*, 151:125–133, 1991.

[10] A. S. Deif. The interval eigenvalue problem. *Z. Angew. Math. Mech.*, 71(1):61–64, 1991.

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

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

[13] G. H. Golub and C. F. Van Loan. *Matrix computations. 3rd ed.* Johns Hopkins University Press, 1996.

[14] E. Hansen and G. W. Walster. *Global optimization using interval analysis. 2nd ed., revised and expanded.* Marcel Dekker, New York, 2004.

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

[16] D. Hertz. Interval analysis: Eigenvalue bounds of interval matrices. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of optimization.*, pages 1689–1696. Springer, New York, 2009.

[17] M. Hladík, D. Daney, and E. Tsigaridas. An algorithm for the real interval eigenvalue problem. Research Report RR-6680, INRIA, France, `http://hal.inria.fr/inria-00329714/en/`, October 2008. sumbitted to J. Comput. Appl. Math.

[18] M. Hladík, D. Daney, and E. Tsigaridas. Bounds on eigenvalues and singular values of interval matrices. Research Report inria-00370603:1, INRIA, France, `http://hal.inria.fr/inria-00370603/en/`, March 2009. sumbitted to SIAM J. Matrix Anal. Appl.

[19] M. Hladík, D. Daney, and E. Tsigaridas. A filtering method for the interval eigenvalue problem. Research Report RR-7057, INRIA, France, `http://hal.inria.fr/inria-00422966/en/`, October 2009. sumbitted to Computing.

[20] R. A. Horn and C. R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1985.

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

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

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

[24] A. Makhorin. GLPK – GNU Linear Programming Kit. `http://www.gnu.org/software/glpk/`.

[25] C. D. Meyer. *Matrix analysis and applied linear algebra.* SIAM, Philadelphia, 2000.

[26] A. Neumaier. *Interval methods for systems of equations.* Cambridge University Press, Cambridge, 1990.

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

[28] Z. Qiu, P. C. Müller, and A. Frommer. An approximate method for the standard interval eigenvalue problem of real non-symmetric interval matrices. *Commun. Numer. Methods Eng.*, 17(4):239–251, 2001.

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

[30] J. Rohn. Interval matrices: Singularity and real eigenvalues. *SIAM J. Matrix Anal. Appl.*, 14(1):82–91, 1993.

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

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

[33] J. Rohn. Solvability of systems of interval linear equations and inequalities. In M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann, editors, *Linear optimization problems with inexact data.*, chapter 2, pages 35–77. Springer, New York, 2006.

[34] J. Rohn and A. Deif. On the range of eigenvalues of an interval matrix. *Comput.*, 47(3-4):373–377, 1992.

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

Contents

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399