# An Eigenvalue Symmetric Matrix Contractor*

Milan Hladík
Charles University, Faculty of Mathematics and Physics, Dept. of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic
milan.hladik@matfyz.cz

Luc Jaulin
DTN (Développement des Technologies Nouvelles), ENSIETA, 2 rue Franccois Verny, 29806 Brest, Cedex 9, France
luc.jaulin@ensta-bretagne.fr

**Abstract**

We propose an eigenvalue contractor for symmetric matrices. Given a symmetric interval matrix $\boldsymbol{A}^S$ and an interval approximation of its eigenvalue sets $\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_n$ the contractor reduces the entries of $\boldsymbol{A}^S$ such that no matrix with eigenvalues in $\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_n$ is omitted. Our contractor is based on sequentially reducing the entries of $\boldsymbol{A}^S$. We discuss properties of the method and demonstrate its performance on examples.

**Keywords:** interval matrix, interval analysis, symmetric matrix, eigenvalue
**AMS subject classifications:** 65G40, 15A18

## 1 Introduction

Many engineering problems can be represented by a *constraint satisfaction problem* (CSP). A CSP consists of

- a set of variables $x_1, \dots, x_n$ which can be Boolean, integers, reals, vectors, matrices, ...,
- a set of constraints $c_1, \dots, c_m$ that should be satisfied by the $x_i$'s, and
- a set of domains $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ which enclose the $x_i$'s.

Most of the time, the domains are intervals [14] but other sets such as ellipsoids, polytopes, etc. could be considered as well [4, 6]. Constraint propagation is an efficient method which is able to contract the domains for the variables in polynomial time without losing any solution [2, 23, 24]. The main idea is to associate each constraint

*Submitted: November 4, 2010; Revised: March 22, 2011; Second revision: June 30, 2011; Third revision: October 4, 2011; Accepted: October 17, 2011.

$c_j$ with an operator $\mathcal{C}_j$ called a *contractor*. $\mathcal{C}_j$ is able to contract the domains of the variables by eliminating values that are not consistent with the constraint $c_j$ [11, 21]. Libraries enclosing contractors for all *primitive constraints* (i.e. constraints that cannot be decomposed such as $y = \sin(x)$ or $z = x + y, \dots$) are available. For a given non-primitive constraint when no contractor is available, the constraint should be decomposed into primitive constraints by introducing new variables. For instance, the constraint $z = x - \sin y$ can be decomposed into the two primitive constraints $z = x - a$ and $a = \sin y$. Such a decomposition introduces pessimism (it is the *local consistency* problem) and should avoided when possible.

A *global constraint* is a non-primitive constraint for which a specific contractor (called a *global contractor*) has been built. A huge catalogue of more than 350 global contractors exists [1]. Most of them are devoted to constraints involving variables that Boolean, integer or real. But only few global contractors exist for constraints involving matrices. Let us quote several contractors related to linear constraints [3] or to the constraint "$A$ is positive semi-definite" [10].

This paper proposes to build a contractor associated with a constraint relating a symmetric matrix with its real eigenvalues. This constraint is of importance for several problems of robot modeling (when an inertia matrix is involved [5]), linear control (see e.g., for linear quadratic control or stability analysis [17, 22]) and parameter identification (via covariance matrices [17]).

The problem considered can be formulated as follows:

$$\left(\begin{array}{cc} \text{(i)} & A \in \boldsymbol{A},\ \ \lambda_1 \in \boldsymbol{\lambda}_1, \dots,\ \lambda_n \in \boldsymbol{\lambda}_n \\ \text{(ii)} & A \text{ is symmetric and } \sigma(A) = \{\lambda_1, \dots, \lambda_n\} \end{array}\right)$$

In line (i), $\boldsymbol{A}$, $\boldsymbol{\lambda}_1, \dots$ , $\boldsymbol{\lambda}_n$ correspond to the interval domains, and the constraint to be treated in the paper is at line (ii); $\sigma(A)$ denotes the spectrum of $A$. A contractor associated with the constraint (ii) is an operator

$$\mathcal{C}\left(\boldsymbol{A}, \boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_n\right) = \left(\boldsymbol{A}', \boldsymbol{\lambda}_1', \dots, \boldsymbol{\lambda}_n'\right)$$

such that

(a) $(\boldsymbol{A}', \boldsymbol{\lambda}_1', \dots, \boldsymbol{\lambda}_n') \subset (\boldsymbol{A}, \boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_n)$

(b) $\left\{\begin{array}{c} A \text{ is symmetric and } \sigma(A) = \{\lambda_1, \dots, \lambda_n\} \\ A \in \boldsymbol{A}, \lambda_1 \in \boldsymbol{\lambda}_1, \dots, \lambda_n \in \boldsymbol{\lambda}_n \end{array}\right. \quad \Rightarrow (A \in \boldsymbol{A}', \lambda_1 \in \boldsymbol{\lambda}_1', \dots, \lambda_n \in \boldsymbol{\lambda}_n')$

(a) is the *contractance* property and (b) is the *correctness* property.

## 2 Preliminaries

For a real symmetric matrix $A \in \mathbb{R}^{n \times n}$ we denote its eigenvalues by $\lambda_1(A) \geq \dots \geq \lambda_n(A)$, and the spectral radius by $\rho(A)$. Let

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

be an interval matrix, where $\underline{A}$, $\overline{A} \in \mathbb{R}^{m \times n}$, $\underline{A} \leq \overline{A}$, are given. We denote the midpoint and radius of $\boldsymbol{A}$ by

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

respectively. A *symmetric interval matrix* is defined as

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

and its eigenvalue sets as

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

where $i = 1, \dots, n$.

The problem of finding a tight enclosure to $\boldsymbol{\lambda}_1(\boldsymbol{A}^S), \dots, \boldsymbol{\lambda}_n(\boldsymbol{A}^S)$ was studied by many authors, see e.g. [8, 18]. Herein, we consider the inverse problem. Let $\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_n$ be given intervals. Our aim is to reduce $\boldsymbol{A}^S$ to a more narrow symmetric interval matrix $\boldsymbol{B}^S \subseteq \boldsymbol{A}^S$ such that we do not lose any matrix with eigenvalues in prescribed intervals. That is, for any $A \in \boldsymbol{A}^S \setminus \boldsymbol{B}^S$ there is $i \in \{1, \dots, n\}$ for which $\lambda_i(A) \notin \boldsymbol{\lambda}_i$.

The problem considered is NP-hard. Moreover, checking whether or not $\boldsymbol{A}^S$ can be reduced to the empty set is also NP-hard; we call it *the existence problem.*

**Theorem 1.** *The existence problem is NP-hard.*

*Proof.* Let $\boldsymbol{M}$ be an interval matrix. It is called *regular* if it contains no singular matrix. Testing regularity of interval matrices is NP-hard problem [15, 18]. So, it is NP-hard to check regularity of the symmetric interval matrix

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

or, equivalently, whether no eigenvalue set of $\boldsymbol{A}^S$ contains zero. Put $\boldsymbol{\lambda}_p = 0$ and $\boldsymbol{\lambda}_i = [-K, K]$, $i \neq p$ and $K > 0$ is large enough. The interval matrix $\boldsymbol{M}$ is regular if and only if $\boldsymbol{A}^S$ can be reduced to the empty set for every $p \in \{1, \dots, n\}$. Hence, if we can solve efficiently a sequence of existence problems for $p = 1, \dots, n$, we can decide on regularity of an interval matrix in polynomial time as well. □

The following theorem recalls the well-known Weyl formulae for the eigenvalues of a matrix sum [9, 25].

**Theorem 2** (Weyl, 1912)**.** *Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices. Then*

$$\begin{aligned} \lambda_{r+s-1}(A+B) &\le \lambda_r(A) + \lambda_s(B) \quad \forall r, s \in \{1, \dots, n\},\ r+s \le n+1, \\ \lambda_{r+s-n}(A+B) &\ge \lambda_r(A) + \lambda_s(B) \quad \forall r, s \in \{1, \dots, n\},\ r+s \ge n+1. \end{aligned}$$

We will also utilize the Rohn theorem [8, 18], which gives simple enclosures for the eigenvalue sets of $\boldsymbol{A}^S$.

**Theorem 3.** *We have*

$$\boldsymbol{\lambda}_i(\boldsymbol{A}^S) \subseteq [\lambda_i(A^c) - \rho(A^\Delta), \lambda_i(A^c) + \rho(A^\Delta)], \quad i = 1, \dots, n.$$

# 3 Method

Without loss of generality, we restrict our consideration to $\boldsymbol{\lambda}_p$, where $p$ is fixed. We try to reduce the upper bound for the $(k, l)$-th and $(l, k)$-th element of $\boldsymbol{A}^S$. Again, $k$ and $l$ remain fixed throughout this section.

For a matrix $M \in \mathbb{R}^{n\times n}$ and a real $r$ denote by $M(r)$ the matrix with elements

$$M(r)_{ij} = \begin{cases} r & \text{if } i=k, j=l \text{ or } i=l, j=k,\\ m_{ij} & \text{otherwise.}\end{cases}$$

Similar notation is used for interval matrices and values.

Our problem is now stated as follows: Find a maximal $\delta > 0$ such that no matrix in $\boldsymbol{A}^S([\overline{a}_{kl}-\delta,\overline{a}_{kl}])$ has its $p$-th eigenvalue in $\boldsymbol{\lambda}_p$. Then, we can contract $\boldsymbol{A}^S$ to $\boldsymbol{A}^S([\underline{a}_{kl},\overline{a}_{kl}-\delta])$.

**Theorem 4.** *Denote*

$$\begin{aligned}
\delta_1 &:= \underline{\lambda}_p - \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)), && n\in\mathbb{N},\ p=1,\dots,n,\\
\delta_2 &:= 2\Big(\underline{\lambda}_p - \lambda_{p-1}(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0))\Big), && n\ge 3,\ p=2,\dots,n,\\
\delta_3 &:= \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \overline{\lambda}_p, && n\in\mathbb{N},\ p=1,\dots,n,\\
\delta_4 &:= 2\Big(\lambda_{p+1}(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \overline{\lambda}_p\Big), && n\ge 3,\ p=1,\dots,n-1,\\
\delta_5 &:= 2\Big(\underline{\lambda}_p - \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0))\Big), && n\ge 2,\ p=1,\dots,n.
\end{aligned}$$

*For non-diagonal entries ($k \neq l$) the values $\delta_1,\delta_2,\delta_3$ and $\delta_4$ can be used for contracting, particularly their maximum*

$$\delta_1^* := \max(\delta_1,\delta_2,\delta_3,\delta_4).$$

*For diagonal entries ($k = l$), we can use*

$$\delta_2^* := \max(\delta_3,\delta_4,\delta_5).$$

*Proof.* First consider the case $k \neq l$. Using Theorem 3, we get

$$\begin{aligned}
\overline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl}-\delta,\overline{a}_{kl}])) &\le \lambda_p(A^c(\overline{a}_{kl}-\tfrac{\delta}{2})) + \rho(A^\Delta(\tfrac{\delta}{2}))\\
&\le \lambda_p(A^c(\overline{a}_{kl}-\tfrac{\delta}{2})) + \rho(A^\Delta(0)) + \rho(0(\tfrac{\delta}{2}))\\
&\le \lambda_p(A^c(\overline{a}_{kl}-\tfrac{\delta}{2})) + \rho(A^\Delta(0)) + \tfrac{\delta}{2}
\end{aligned}$$

Using the Weyl theorem, we can obtain two upper bounds for the first term in the right-hand side. First,

$$\begin{aligned}
\lambda_p(A^c(\overline{a}_{kl}-\tfrac{\delta}{2})) &\le \lambda_p(A^c(\overline{a}_{kl})) + \lambda_1(0(-\tfrac{\delta}{2}))\\
&\le \lambda_p(A^c(\overline{a}_{kl})) + \tfrac{\delta}{2}
\end{aligned}$$

Combining together,

$$\begin{aligned}
\overline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl}-\delta,\overline{a}_{kl}])) &\le \lambda_p(A^c(\overline{a}_{kl})) + \tfrac{\delta}{2} + \rho(A^\Delta(0)) + \tfrac{\delta}{2}\\
&\le \lambda_p(A^c(\overline{a}_{kl})) + \rho(A^\Delta(0)) + \delta.
\end{aligned}$$

Since we want not to exceed the bound $\underline{\lambda}_p$, we have

$$\lambda_p(A^c(\overline{a}_{kl})) + \rho(A^\Delta(0)) + \delta \le \underline{\lambda}_p,$$

whence

$$\delta \leq \underline{\lambda}_p - \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) = \delta_1.$$

Now we use another inequality of the Weyl theorem. For $p \geq 2$ and $n \geq 3$ we have

$$\begin{aligned}\lambda_p(A^c(\overline{a}_{kl} - \tfrac{\delta}{2})) &\leq \lambda_{p-1}(A^c(\overline{a}_{kl})) + \lambda_2(0(-\tfrac{\delta}{2}))\\ &\leq \lambda_{p-1}(A^c(\overline{a}_{kl})) + 0.\end{aligned}$$

Hence,

$$\overline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl} - \delta, \overline{a}_{kl}])) \leq \lambda_{p-1}(A^c(\overline{a}_{kl})) + \rho(A^\Delta(0)) + \tfrac{\delta}{2} \leq \underline{\lambda}_p,$$

and thus

$$\delta \leq 2\Big(\underline{\lambda}_p - \lambda_{p-1}(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0))\Big) = \delta_2.$$

Similarly, in order not to exceed the upper bound $\overline{\lambda}_p$ we obtain $\delta_3$ and $\delta_4$. By Theorem 3,

$$\underline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl} - \delta, \overline{a}_{kl}])) \geq \lambda_p(A^c(\overline{a}_{kl} - \tfrac{\delta}{2})) - \rho(A^\Delta(0)) - \tfrac{\delta}{2}.$$

Using the Weyl theorem,

$$\begin{aligned}\lambda_p(A^c(\overline{a}_{kl} - \tfrac{\delta}{2})) &\geq \lambda_p(A^c(\overline{a}_{kl})) + \lambda_n(0(-\tfrac{\delta}{2}))\\ &\geq \lambda_p(A^c(\overline{a}_{kl})) - \tfrac{\delta}{2},\end{aligned}$$

whence

$$\underline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl} - \delta, \overline{a}_{kl}])) \geq \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \delta.$$

To have

$$\lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \delta \geq \overline{\lambda}_p,$$

we must have

$$\delta \leq \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \overline{\lambda}_p = \delta_3.$$

Using another Weyl inequality, we have

$$\begin{aligned}\lambda_p(A^c(\overline{a}_{kl} - \tfrac{\delta}{2})) &\geq \lambda_{p+1}(A^c(\overline{a}_{kl})) + \lambda_{n-1}(0(-\tfrac{\delta}{2}))\\ &\geq \lambda_{p+1}(A^c(\overline{a}_{kl})) + 0.\end{aligned}$$

Therefore,

$$\underline{\lambda}_p(\boldsymbol{A}^S([\overline{a}_{kl} - \delta, \overline{a}_{kl}])) \geq \lambda_{p+1}(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \tfrac{\delta}{2} \geq \overline{\lambda}_p,$$

whence

$$\delta \leq 2\Big(\lambda_{p+1}(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) - \overline{\lambda}_p\Big) = \delta_4$$

for the second one.

The above considerations hold true for the case $k = l$, too, but better results can be obtained. Since $\lambda_1(0(-\frac{\delta}{2})) = 0$, in the derivation of $\delta_1$ we obtain

$$\lambda_p(A^c(\overline{a}_{kl})) + \rho(A^\Delta(0)) + \tfrac{\delta}{2} \leq \underline{\lambda}_p.$$

So

$$\delta \leq 2\Big(\underline{\lambda}_p - \lambda_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0))\Big) = \delta_5.$$

The value $\delta_2$ is out of consideration because $\delta_2 \leq \delta_5$ always holds. The remaining $\delta_3$ and $\delta_4$ can be utilized as they are. □

In the proof we utilized only two inequalities of the Weyl theorem. Note that the remaining ones cannot improve the results (with exception of the last one, which is improbable).

Note also that some of the values $\delta_1, \delta_2, \delta_3, \delta_4$ and $\delta_5$ may be negative. Typically, one of the pair $\delta_1$ and $\delta_3$, and one of the pair $\delta_2$ and $\delta_4$ is negative.

The method based on Theorem 4 is straightforward and is described in Algorithm 1. For contracting entries of $\boldsymbol{A}^S$ from below we can develop similar formulae or realize that eigenvalues of $-A$ are negatives of eigenvalues of $A$. That is, instead of contracting $\boldsymbol{A}^S$ from below we contract $-\boldsymbol{A}^S$ from above. Note that a result of this inversion is, in step 6 of the algorithm, the index of the interval $-\boldsymbol{\lambda}_p$ is $n + 1 - p$.

**Algorithm 1** (Eigenvalue symmetric matrix contractor)

1: **while** improvement is significant **do**
2:   **for** $k = 1, \dots, n$ **do**
3:     **for** $l = k + 1, \dots, n$ **do**
4:       compute $\delta_1^*$ according to Theorem 4 for $p = 1, \dots, n$;
5:       $\boldsymbol{A}^S := \boldsymbol{A}^S([\underline{a}_{kl}, \overline{a}_{kl} - \delta_1^*])$;
6:       compute $\delta^*$ according to Theorem 4 for the matrix $-\boldsymbol{A}^S$ and interval $-\boldsymbol{\lambda}_p$ $p = 1, \dots, n$;
7:       $\boldsymbol{A}^S := \boldsymbol{A}^S([\underline{a}_{kl} + \delta_1^*, \overline{a}_{kl}])$;
8:     **end for**
9:     compute $\delta_2^*$ according to Theorem 4 for $p = 1, \dots, n$;
10:     $\boldsymbol{A}^S := \boldsymbol{A}^S([\underline{a}_{kk}, \overline{a}_{kk} - \delta_2^*])$;
11:     compute $\delta_2^*$ according to Theorem 4 for the matrix $-\boldsymbol{A}^S$ and interval $-\boldsymbol{\lambda}_p$, $p = 1, \dots, n$;
12:     $\boldsymbol{A}^S := \boldsymbol{A}^S([\underline{a}_{kk} + \delta_2^*, \overline{a}_{kk}])$;
13:   **end for**
14: **end while**
15: **return** $\boldsymbol{A}^S$;

# 4 Properties of the contractor

Provided we limit the number of loops of Algorithm 1 by a polynomially large constant, the whole algorithm runs in polynomial time. Specifically, the algorithm has complexity $\mathcal{O}(In^5)$, where $I$ is the number of iterations. The factor $\mathcal{O}(n^2)$ corresponds to the number of elements in $\boldsymbol{A}^S$ to be processed. For each element of $\boldsymbol{A}^S$, we have to calculate all eigenvalues and the spectral radius of a point matrix, which costs $\mathcal{O}(n^3)$. Since we use the same point matrix for each $p = 1, \ldots, n$, the total cost for one element of $\boldsymbol{A}^S$ is $\mathcal{O}(n^3)$.

The contractor does not converge to optimal bounds in general. This is because the estimations used in the proof of Theorem 4 needn't be very sharp, and also we cannot expect our algorithm to efficiently solve a problem that is NP-hard.

Nevertheless, our contractor is so called *thin* [11] (cf. notion of thin inclusion function [7, 14]). This means, in the case there is no solution to the problem ($\boldsymbol{A}^S$ is reducible to the empty set), we can conclude that there is no solution, provided that the interval matrix is narrow enough.

Let $\|A^\Delta\|_2 \leq \varepsilon$, where $\varepsilon > 0$ and $\|\cdot\|_2$ denotes the matrix 2-norm. Suppose that for some $p \in \{1, \ldots, n\}$ one has $\overline{\lambda}_p(\boldsymbol{A}^S) < \underline{\lambda}_p$, that is, the problem has no solution. Denoting $c := \underline{\lambda}_p - \overline{\lambda}_p(\boldsymbol{A}^S)$, one reads

$$\begin{aligned}\delta_1 &= \underline{\lambda}_p - \overline{\lambda}_p(A^c(\overline{a}_{kl})) - \rho(A^\Delta(0)) \\ &\geq \underline{\lambda}_p - \overline{\lambda}_p(\boldsymbol{A}^S) - \rho(A^\Delta(0)) \\ &\geq c - \varepsilon.\end{aligned}$$

We used the fact that $\rho(A^\Delta(0)) = \|A^\Delta(0)\|_2 \leq \|A^\Delta\|_2$, which is true due to symmetry and non-negativity of those matrices [9, 13].

As long as $c > \varepsilon$ we can conclude that there is no solution in at most $\lceil \frac{\varepsilon}{c-\varepsilon} \rceil$ iterations of the contractor. Thus, if $\boldsymbol{\lambda}_p$ and $\boldsymbol{\lambda}_p(\boldsymbol{A}^S)$ are disjoint for some $p \in \{1, \ldots, n\}$, and the entries of $\boldsymbol{A}^S$ are narrow enough, we can verify non-existence in a finite number of steps.

# 5 Numerical experiments

**Example 1.** Consider a symmetric interval matrix [8, 16, 26]

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

It is known [8] that the optimal eigenvalue sets are

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

Let intervals

$$\begin{aligned}\boldsymbol{\lambda}_1 &= [12710, 12730] & \boldsymbol{\lambda}_2 &= [7000, 7130], \\ \boldsymbol{\lambda}_3 &= [3300, 3450], & \boldsymbol{\lambda}_4 &= [800, 850].\end{aligned}$$

be given. We will contract entries of $\boldsymbol{A}^S$ according to these intervals. The first iteration of Algorithm 1 reduces the matrix to

$$\begin{pmatrix} [2975, 3025] & [-2015, -1985] & 0 & 0 \\ [-2015, -1985] & [4965, 5035] & [-3020, -2980.9] & 0 \\ 0 & [-3020, -2980.9] & [6971.4, 7041.9] & [-4025, -4004.6] \\ 0 & 0 & [-4025, -4004.6] & [8966.5, 9055] \end{pmatrix}^S.$$

The next iterations yields

$$\begin{pmatrix} [2975, 3024.2] & [-2015, -1991.3] & 0 & 0 \\ [-2015, -1991.3] & [4965, 5026.2] & [-3020, -2995.6] & 0 \\ 0 & [-3020, -2995.6] & [6982.4, 7035.8] & [-4025, -4011] \\ 0 & 0 & [-4025, -4011] & [8984.4, 9055] \end{pmatrix}^S,$$

and the third one results in

$$\begin{pmatrix} [2975, 3017.9] & [-2015, -2002.2] & 0 & 0 \\ [-2015, -2002.2] & [4965, 5016.1] & [-3020, -3005.4] & 0 \\ 0 & [-3020, -3005.4] & [6990.8, 7030.9] & [-4025, -4019.2] \\ 0 & 0 & [-4025, -4019.2] & [9000.4, 9053] \end{pmatrix}^S.$$

Finally, in the fourth iteration it turns out that the interval matrix is reduced to the empty set. The $(3, 4)$-th element of $\boldsymbol{A}^S$ is contracted from above to $[-4025, -4021.5]$ and from below to the empty set, since the corresponding contracting subtrahends are $\delta_1^* = 2.7117$ for $p = 1$, zero for $p = 2$ and $p = 3$, and $\delta_1^* = 0.7992$ for $p = 4$. We conclude that there is no symmetric matrix $A \in \boldsymbol{A}^S$ whose eigenvalues lie inside $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, $\boldsymbol{\lambda}_3$ and $\boldsymbol{\lambda}_4$, respectively.

This example shows the strength of Algorithm 1, because we were able to conclude non-existence even though the intervals $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_i(\boldsymbol{A}^S)$ are non-disjoint for every $i = 1, \dots, 4$.

**Example 2.** Now, we compare our approach with a simple bisection method introduced in [12] as a 3-B (or shaving) method. The bisection method sequentially goes through the interval matrix entries, and each of them splits into two intervals. Thus, for each $k, l = 1, \dots, n$ we determine $\boldsymbol{A}^S([\underline{a}_{kl}, a^c_{kl}])$ and $\boldsymbol{A}^S([a^c_{kl}, \overline{a}_{kl}])$. We calculate the eigenvalue set enclosures $\boldsymbol{\lambda}'_1, \dots, \boldsymbol{\lambda}'_n$ by Theorem 3 for each of these two matrices. If

$$\boldsymbol{\lambda}_i \cap \boldsymbol{\lambda}'_i = \emptyset$$

for some $i$, we can reduce the interval $\boldsymbol{a}_{kl}$ to $[a^c_{kl}, \overline{a}_{kl}]$ or $[\underline{a}_{kl}, a^c_{kl}]$, respectively. Since the splitting is sequential, it needs just $\frac{1}{2}n(n-1)$ splittings, which makes the method tractable.

For the comparisons, we considered randomly generated matrices of dimensions $n \in \{5, 10, 15, 20, 25, 50\}$. The matrices $\boldsymbol{A}^S$ were generated as follows. The entries of $A^c$ were chosen randomly from $[-10, 10]$ and the entries of $A^\Delta$ from $[0, 1]$ with uniform distribution. We calculated eigenvalue enclosures $\boldsymbol{\lambda}_i$, $i = 1, \dots, n$ by Theorem 3. Then, we put $\boldsymbol{\lambda}_i(r) = [\underline{\lambda}_i, \underline{\lambda}_i + 2r\lambda^\Delta]$, $i = 1, \dots, n$ with parameter $r$. We determined the maximal $r$ such that the corresponding method is able to conclude that there is no solution. We denote the maximal values by $r_1$ and $r_2$ for the bisection method and our approach, respectively.

The experiments were carried out with Matlab 7.11.0 (R2010b) with the support of the packages INTLAB 6 (see [20]) and VERSOFT 10 (see [19]). INTLAB 6 provides interval arithmetic and useful interval functions, and VERSOFT 10 collects verification functions. (We used it for verified eigenvalue computations.) Thus, all of our computations were reliable and the results are verified. Tables 1 and 2 show the results. In Table 1, we display ratios $\frac{r_1}{r_2}$ for various dimensions; the ratios are the means of several runs. It shows that our approach is able to verify non-existence for about 40% to 60% wider intervals then the simple bisection.

| $n$ | 5 | 10 | 15 | 20 | 25 | 50 |
|---|---|---|---|---|---|---|
| $\frac{r_1}{r_2}$ | 0.6542 | 0.5820 | 0.6311 | 0.7444 | 0.7677 | 0.7516 |

Table 1: Efficiency comparison of our approach and the bisection method.

Table 2 presents running times in seconds for three cases. Case I stands for the situation when both methods fail to prove non-existence, i.e. $r$ is too large, whereas case II denotes the situation when both methods succeed at verifying non-existence. Case III covers the remaining situation when our method succeed and bisection fails. The table shows that our method is slightly slower in case I, but it is faster in case II up to dimension 20. In case III our method is several times slower for small dimensions, however, contrary to the bisection method it proves non-existence.

| $n$ | case I | | case II | | case III | |
|---|---|---|---|---|---|---|
| | our method | bisection | our method | bisection | our method | bisection |
| 5 | 4.7235 | 3.5637 | 1.2699 | 1.7595 | 40.105 | 3.5648 |
| 10 | 43.627 | 40.176 | 6.6407 | 9.2194 | 156.03 | 40.285 |
| 15 | 187.18 | 176.55 | 25.384 | 30.580 | 530.30 | 176.85 |
| 20 | 546.63 | 520.46 | 40.202 | 101.63 | 1511.7 | 520.58 |
| 25 | 1255.3 | 1209.5 | 90.962 | 70.701 | 1647.9 | 1211.7 |
| 50 | 18359 | 17994 | 273.83 | 98.809 | 17959 | 18128 |

Table 2: Time comparison of our approach and the bisection method.

# 6 Conclusion

We have proposed a method for contracting entries of interval matrices and thus removing the redundant subsets that have no eigenvalue in a priori given eigenvalue estimates. We have carried out numerical experiments and comparisons with a simple bisection method. Provided the eigenvalue estimates are very tiny or very wide, the bisection method is often slightly faster than ours. On the other hand, our approach can solve the difficult cases and prove non-existence for wider intervals than the bisection method.

In future work, further analysis of the structure of the problem will yield improvements.

## References


[1] Nicolas Beldiceanu, Mats Carlsson, and Jean-Xavier Rampon. *Global Constraint Catalog*. University of Nantes, 2009. Available at `http://www.emn.fr/x-info/sdemasse/gccat/`.

[2] Christian Bliek, Peter Spellucci, Luís N. Vincente, Arnold Neumaier, Laurent Granvilliers, Etienne Huens, Pascal Van Hentenryck, Djamila Sam-Haroud, and Boi Faltings. Algorithms for Solving Nonlinear Constrained and Optimisation Problems: State of the Art. A 222 page progress report of the COCONUT project, available at `http://www.mat.univie.ac.at/~neum/glopt/coconut/cocbib.html`, 2001.

[3] Gilles Chabert and Luc Jaulin. QUIMPER, a language for quick interval modelling and programming in a bounded-error context. *Artif. Intell.*, 173:1079–1100, 2009.

[4] C. Combastel. A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes. In *44th IEEE Conference on Decision and Control & European Control Conference CDC-ECC'05*, pages 7228–7234, 2005.

[5] Etienne Dombre and Wisama Khalil. *Robot manipulators: modeling, performance analysis and control*. Wiley-ISTE, London, 2007.

[6] C. Durieu, B. Polyak, and E. Walter. Ellipsoidal state outer-bounding for MIMO systems via analytical techniques. In *Proceedings of the IMACS—IEEE–SMC CESA'96 Symposium on Modelling and Simulation, volume 2*, pages 843–848, 1996.

[7] E. R. Hansen. Bounding the solution of interval linear equations. *SIAM J. Numer. Anal.*, 29(5):1493–1503, 1992.

[8] Milan Hladík, David Daney, and Elias Tsigaridas. Bounds on real eigenvalues and singular values of interval matrices. *SIAM J. Matrix Anal. Appl.*, 31(4):2116–2129, 2010.

[9] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1985.

[10] Luc Jaulin and Didier Henrion. Contracting optimally an interval matrix without loosing any positive semi-definite matrix is a tractable problem. *Reliab. Comput.*, 11(1):1–17, 2005.

[11] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Applied interval analysis. With examples in parameter and state estimation, robust control and robotics.* Springer, London, 2001.

[12] Olivier Lhomme. Consistency techniques for numeric CSPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 232–238, Chambéry, France, 1993.

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

[14] Ramon E. Moore. *Methods and applications of interval analysis.* SIAM, Philadelphia, PA, 1979.

[15] Svatopluk Poljak and Jiří Rohn. Checking robust nonsingularity is NP-hard. *Math. Control Signals Syst.*, 6(1):1–9, 1993.

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

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

[18] Jiří Rohn. A handbook of results on interval linear problems. available at `http://www.cs.cas.cz/rohn/handbook`, 2005.

[19] Jiří Rohn. VERSOFT: Verification software in MATLAB / INTLAB, version 10, 2009. `http://uivtx.cs.cas.cz/~rohn/matlab/`.

[20] Siegfried M. Rump. INTLAB – INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tu-harburg.de/rump/`.

[21] Djamila Sam-Haroud. *Constraint consistency techniques for continuous domains.* PhD thesis, Swiss Federal Institute of Technology in Lausanne, Switzerland, 1995. PhD dissertation No. 1423.

[22] C. Scherer and S. Weiland. Course on LMIs in Control. Lecture Notes at Delft University of Technology and Eindhoven University of Technology, 2002.

[23] M. H. van Emden. Algorithmic power from declarative use of redundant constraints. *Constraints*, 4(4):363–381, 1999.

[24] Pascal van Hentenryck, Laurent Michel, and Yves Deville. *Numerica – A Modelling Language for Global Optimization.* MIT Press, Cambridge, Massachusetts, 1997.

[25] James Hardy Wilkinson. *The algebraic eigenvalue problem. 1. paperback ed.* Clarendon Press, Oxford, 1988.

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